

Temporal Logics, Automata and the Modal μ -Calculus

MSc Thesis (*Afstudeerscriptie*)

written by

Sander in 't Veld

(born October 13th, 1993 in Haarlem)

under the supervision of **prof. dr. Yde Venema** and **dr. Sebastian Enqvist**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense: **Members of the Thesis Committee:**
July 4th, 2016

dr. Alexandru Baltag
dr. Facundo Carreiro
dr. Sebastian Enqvist
dr. Helle Hansen
dr. Jakub Szymanik (chair)
prof. dr. Yde Venema



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

Computation tree logic (CTL) and its extension CTL* offer a rigorous approach to program verification. The highly expressive modal μ -calculus subsumes both CTL and CTL* while remaining computationally well-behaved. Translations from CTL and CTL* into the modal μ -calculus are known, but the resulting fragments have not been identified syntactically. Having an exact characterization of a logic as a fragment of the modal μ -calculus gives a better understanding of the expressivity of both logics involved. An automata theoretic approach serves to form a bridge between logics and game semantics are instrumental when comparing formulas with automata.

In this thesis CTL* is translated into a class of modal parity automata. An exact characterization of this class of automata as a fragment of the modal μ -calculus is given. Furthermore CTL is fully characterized both as a class of modal automata with singleton clusters and as a one-variable fragment of the modal μ -calculus.

Keywords: computation tree logic, game semantics, automata theory, modal μ -calculus.

Contents

1	Introduction	2
2	Preliminaries	5
2.1	The logics CTL* and CTL	6
2.2	Board games	8
2.3	The modal μ -calculus	10
2.4	Stream automata	14
2.5	Modal automata	14
3	Dominance in evaluation games	19
3.1	Evaluation games for CTL*	19
3.2	Dominated formulas	25
3.3	Cluster-path acceptance	27
4	An automaton construction for CTL*	32
4.1	Automaton construction	32
4.2	Automaton modification	42
5	Dominance in the modal μ-calculus and its automata	48
5.1	Automata and the modal μ -calculus	48
5.2	Dominated fixpoints	53
5.3	Mostowski index and alternation depth	56
6	A characterization of CTL	61
6.1	Automata for CTL	61
6.2	CTL as a fragment of the modal μ -calculus	67
7	Conclusion	69
	Bibliography	72
	Index	74

Chapter 1

Introduction

Debugging – the task of identifying and resolving flaws in a computer program – is a standard chore for any programmer. For small programs a combination of reasoning and testing is enough to solve most issues, but as programs grow more complex they become less transparent and the need for rigorous debugging methods increases. For concurrent systems in particular it can be essential to ensure various liveness and safety properties in order to avoid unforeseen deadlock or starvation. Work done by Lamport (1977) and others show that mathematical proofs can aid in this respect.

The field of logic can formalize this method of program verification even further. Temporal properties can be described using the temporal operators **F** and **G** from Prior (1957), which state that something holds at some point in the future and at every point in the future respectively. Pnueli (1977) introduced these operators to the world of program verification and showed that linear temporal logic (LTL) can be used to encode liveness and safety. For instance, the property that every request (r) made at any time during the program execution is eventually satisfied (s) can be expressed by the formula $\mathbf{G}(r \rightarrow \mathbf{F}s)$. The use of LTL is limited by its linear nature, since programs often rely on branching. If one models the execution of a program as a computation tree, then LTL can only be used to describe the behavior along a single path down the tree at a time. Computation tree logic (CTL) as introduced by Clarke and Emerson (1981) is capable of describing properties of nodes in a tree with respect to different paths simultaneously. It has operators such as **AF** and **EF**, which state that something will hold eventually in every possible future and in some possible future respectively. However CTL has the opposite problem, in that it cannot express some interesting properties of individual paths, such as $\mathbf{G}(r \rightarrow \mathbf{F}s)$. In Emerson and Halpern (1986) both approaches are unified in a logic called CTL*. Its key feature is that the operators **E** and **A** are used to quantify an LTL formula over individual paths originating from a single node. Its syntax separates ‘state-formulas’ which are evaluated at the nodes of a tree from ‘path-formulas’ which are evaluated at the paths. This separation is somewhat similar to the way propositional dynamic logic (PDL) is defined in terms of formulas and programs.

Another logic used in verification is the modal μ -calculus (μ ML), which was described in its current form by Kozen (1983). The logic μ ML is very expressive and at the same time

computationally well-behaved. However its formulas are less human-readable than those from temporal and dynamic logics such as LTL, CTL and PDL. It is therefore worthwhile to find translations from these logics into μML . The operators of CTL were immediately shown to be expressible by fixpoints in Clarke and Emerson (1981), and Dam (1990) gives a direct translation of CTL* into the modal μ -calculus. A translation from PDL into μML was given by Pratt (1981). Translations back to CTL*, PDL etcetera are generally impossible because these logics are not expressive enough to capture the full power of μML . At the same time, knowing exactly which fragment of μML a logic corresponds to can give a better understanding of the expressivity of both logics involved. A characterization of PDL as a fragment of μML is due to Carreiro and Venema (2014). Characterizations for LTL, CTL and CTL* in μML do not yet exist, but these logics have been characterized as second-order logics. Kamp (1968) shows that LTL corresponds to first order logic on linear models, and Moller and Rabinovich (2001) show that CTL* corresponds to the bisimulation invariant fragment of monadic path logic – the fragment of monadic second order logic where quantification over sets is restricted to sets that are path-like. This result relates back to μML in the context of the work by Janin and Walukiewicz (1996), where the modal μ -calculus is shown to be as expressive as the bisimulation invariant fragment of monadic second order logic.

Janin and Walukiewicz use automata as an intermediary between the modal μ -calculus and monadic second order logic. In Janin and Walukiewicz (1995) it is shown that the properties expressible in μML are exactly those properties that are definable by FO-automata – that is, automata whose transition terms are formulas in first order logic without equality. Similarly in Walukiewicz (1996) monadic second order logic is shown to be equivalent over trees to FOE-automata – automata over first order logic with equality. Then Janin and Walukiewicz (1996) show that these two classes of automata are equivalent under bisimulation invariance. Comparing logical formulas with automata is somewhat similar to Kleene’s Theorem, due to Kleene (1951), which states that regular languages are exactly those languages definable by finite automata. The class of automata that corresponds to PDL is given by Carreiro and Venema (2014). Diekert and Gastin (2008) show that LTL corresponds to counter-free Büchi stream automata.

When comparing formulas with automata, it proves convenient to take a game theoretical perspective on truth. Using game semantics to prove and disprove properties is used to great effect in the Ehrenfeucht-Fraïssé games, due to Fraïssé (1955) and Ehrenfeucht (1961). *Evaluation games* are two-player games where one player tries to verify a formula and the other tries to falsify it. For example when evaluating a formula of the form $\varphi \vee \psi$, it is the verifier who chooses to verify either φ or ψ ; conversely for a formula $\varphi \wedge \psi$ both conjuncts must hold, so the falsifier may decide with which subformula the game continues. Automata have *acceptance games* where a verifier tries to show that a word, stream or system is accepted by an automaton and a falsifier tries to have the automaton reject it. Especially when looking at modal automata – automata whose transition terms are modal formulas – the two games are very similar.

This thesis introduces a class of modal parity automata that CTL* formulas can be translated into, and gives a characterization of this class as a syntactic fragment of the

modal μ -calculus. A main result is that the modal automata in this class and the μ ML formulas that correspond to them make use of modal formulas that are *dominated* by one of the two players; roughly speaking, in the evaluation match of a dominated formula one player can control the flow of the game whereas the other player has relatively little power. A third result is that CTL can be characterized by restricting this class of automata to automata with singleton clusters, and consequently restricting the fragment of μ ML to be one-variable.

The next chapter lays the groundwork on CTL*, board games, the modal μ -calculus and modal automata. In chapter 3, evaluation game semantics for CTL* are given and the syntactic property of modal formulas called “dominance” is introduced. A construction of modal automata for CTL* formulas is presented in chapter 4. The problem of a translation back into CTL* is discussed briefly in chapter 5, where the class of automata with dominated clusters is characterized in μ ML. In chapter 6 a characterization of CTL is given, both as a class of modal automata and as a fragment of μ ML. Finally, chapter 7 will contain a summary of our results, discuss the translation from automata back to CTL* and pose a number of open questions for future work.

Chapter 2

Preliminaries

Literals will serve as the smallest elements of all the logics and automata featured in this paper. Throughout this paper we will assume that Prop is a finite set of proposition letters. Other than proposition letters from Prop and their negations, we have the constants \top for truth and \perp for falsehood.

Definition 2.1. The set Lit of *literals* over Prop is generated by

$$\ell ::= \top \mid \perp \mid p \mid \neg p$$

where $p \in \text{Prop}$. ◁

We will consistently use the symbol ℓ to refer to elements of Lit . Literals are given their meaning by valuations in the standard manner.

Definition 2.2. Let $V : \text{Prop} \rightarrow \wp(S)$ be a valuation of Prop on a set S of states. Define satisfaction of a literal ℓ at a state $s \in S$ under the valuation V , denoted $s \Vdash_V \ell$, by

$s \Vdash_V \top$	always
$s \Vdash_V \perp$	never
$s \Vdash_V p$	if $s \in V(p)$
$s \Vdash_V \neg p$	if $s \notin V(p)$

where $p \in \text{Prop}$. ◁

The structures considered in this paper are valued transition systems over Prop , i.e. monomodal Kripke models with serial accessibility relations. Note that infinite trees can be seen as a special case of a serial transition system.

Definition 2.3. A *transition system* is $\mathbb{S} = (S, R, V)$ where S is a set of states, $R \subseteq S \times S$ a serial relation and $V : \text{Prop} \rightarrow \wp(S)$ a valuation. ◁

Seriality means that for every $s \in S$ there is $t \in S$ with sRt . As a result one can construct, from any starting state $s \in S$, an infinite path that goes from state to state via the accessibility relation R .

Definition 2.4. Let \mathbb{S} be a transition system. A *path* through a transition system \mathbb{S} is a map $\pi : \mathbb{N} \rightarrow S$ such that $\pi(i)R\pi(i+1)$ for all $i \in \mathbb{N}$. For a path π and $k \in \mathbb{N}$, define the path π^k by $\pi^k(i) := \pi(k+i)$ for all $i \in \mathbb{N}$. Let $\Pi(\mathbb{S})$ denote the set of all paths through a transition system \mathbb{S} , and let $\Pi(\mathbb{S}, s)$ denote the set of all paths through \mathbb{S} with $\pi(0) = s$. \triangleleft

2.1 The logics CTL* and CTL

Traditionally, the formulas of CTL* can be generated by the grammar

$$\begin{aligned}\varphi &::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{E}\psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi\end{aligned}$$

where $p \in \text{Prop}$, where \mathbf{E} is the existential path-quantifier, \mathbf{X} is the ‘next time’ operator and \mathbf{U} is the ‘until’ operator. However as we work with evaluation games for CTL*, it is more convenient to work in a negation normal form, so that negations only occur at the level of proposition letters.

Definition 2.5. The syntax of CTL* is generated by the dual grammar

$$\begin{aligned}\varphi &::= \ell \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\psi \mid \mathbf{A}\psi \\ \psi &::= \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi \mid \psi\mathbf{R}\psi\end{aligned}$$

where $\ell \in \text{Lit}$. The set CTL_{Σ}^* of state-formulas of CTL* is generated by φ . The set CTL_{Π}^* of path-formulas of CTL* is generated by ψ . \triangleleft

The operator \mathbf{R} , also known as ‘release’, is the dual of \mathbf{U} . When $\varphi\mathbf{U}\psi$ is taken to mean “ ψ will hold at some point in the future and until that time φ holds”, $\varphi\mathbf{R}\psi$ can be explained as “ ψ will hold forever in the future, unless it is at some point released by φ ”. Note that \mathbf{X} is its own dual. Two other common temporal operators, \mathbf{F} and \mathbf{G} , can be defined by $\mathbf{F}\varphi := \top\mathbf{U}\varphi$ and $\mathbf{G}\varphi := \perp\mathbf{R}\varphi$ respectively. We separate state-formulas, which are evaluated on states, from path-formulas, which are evaluation on paths. However this separation is not at all strict, as every state-formula is also a path-formula.

Historically the logic CTL* is an extension of CTL, and the logic CTL was presented with the operators \mathbf{EX} , \mathbf{AU} and \mathbf{EU} as atomic elements of the language. In this paper we treat CTL as a fragment of CTL* by restricting the shapes of the path-formulas. The semantics of a CTL formula will therefore be given by the semantics of that formula when seen as a CTL* formula.

Definition 2.6. The syntax of CTL is generated by the dual grammar

$$\begin{aligned}\varphi &::= \ell \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\psi \mid \mathbf{A}\psi \\ \psi &::= \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{R}\varphi\end{aligned}$$

where $\ell \in \text{Lit}$. The set CTL_{Σ} of state-formulas of CTL is generated by φ . The set CTL_{Π} of path-formulas of CTL is generated by ψ . \triangleleft

The relational semantics for CTL* are entirely standard.

Definition 2.7. Let \mathbb{S} be a transition system. Define satisfaction of a formula $\varphi \in \text{CTL}_\Sigma^*$ at a state $s \in S$, denoted $\mathbb{S}, s \Vdash \varphi$, and of a formula $\psi \in \text{CTL}_\Pi^*$ at a path $\pi \in \Pi(\mathbb{S})$, denoted $\mathbb{S}, \pi \Vdash \psi$, as follows:

$\mathbb{S}, s \Vdash \ell$	$\Leftrightarrow s \Vdash_V \ell$
$\mathbb{S}, s \Vdash \varphi_1 \vee \varphi_2$	$\Leftrightarrow \mathbb{S}, s \Vdash \varphi_1$ or $\mathbb{S}, s \Vdash \varphi_2$
$\mathbb{S}, s \Vdash \varphi_1 \wedge \varphi_2$	$\Leftrightarrow \mathbb{S}, s \Vdash \varphi_1$ and $\mathbb{S}, s \Vdash \varphi_2$
$\mathbb{S}, s \Vdash \mathbf{E}\psi$	$\Leftrightarrow \exists \pi \in \Pi(\mathbb{S}, s) (\mathbb{S}, \pi \Vdash \psi)$
$\mathbb{S}, s \Vdash \mathbf{A}\psi$	$\Leftrightarrow \forall \pi \in \Pi(\mathbb{S}, s) (\mathbb{S}, \pi \Vdash \psi)$
$\mathbb{S}, \pi \Vdash \varphi$	$\Leftrightarrow \mathbb{S}, \pi(0) \Vdash \varphi$ whenever $\varphi \in \text{CTL}_\Sigma^*$
$\mathbb{S}, \pi \Vdash \psi_1 \vee \psi_2$	$\Leftrightarrow \mathbb{S}, \pi \Vdash \psi_1$ or $\mathbb{S}, \pi \Vdash \psi_2$
$\mathbb{S}, \pi \Vdash \psi_1 \wedge \psi_2$	$\Leftrightarrow \mathbb{S}, \pi \Vdash \psi_1$ and $\mathbb{S}, \pi \Vdash \psi_2$
$\mathbb{S}, \pi \Vdash \mathbf{X}\psi$	$\Leftrightarrow \mathbb{S}, \pi^1 \Vdash \psi$
$\mathbb{S}, \pi \Vdash \psi_1 \mathbf{U} \psi_2$	$\Leftrightarrow \exists k (\mathbb{S}, \pi^k \Vdash \psi_2 \text{ and } \forall i < k (\mathbb{S}, \pi^i \Vdash \psi_1))$
$\mathbb{S}, \pi \Vdash \psi_1 \mathbf{R} \psi_2$	$\Leftrightarrow \forall k (\mathbb{S}, \pi^k \Vdash \psi_2 \text{ or } \exists i < k (\mathbb{S}, \pi^i \Vdash \psi_1))$

where $\ell \in \text{Lit}$. ◁

If φ and ψ are state-formulas we will write $\varphi \equiv \psi$ whenever φ and ψ are logically equivalent, i.e. when $\mathbb{S}, s \Vdash \varphi$ iff $\mathbb{S}, s \Vdash \psi$ for all transition systems \mathbb{S} and all $s \in S$. We will also use this notation for path-formulas.

As said, we define CTL* in a negation normal form. This is not a loss in expressivity, since logical complementation can still be defined.

Definition 2.8. Define the *logical complement* $\neg\varphi$ of a CTL* formula φ by

$\neg\top := \perp$	$\neg\perp := \top$
$\neg(p) := \neg p$	$\neg(\neg p) := p$
$\neg(\varphi \vee \psi) := \neg\varphi \wedge \neg\psi$	$\neg(\varphi \wedge \psi) := \neg\varphi \vee \neg\psi$
$\neg\mathbf{E}\varphi := \mathbf{A}\neg\varphi$	$\neg\mathbf{A}\varphi := \mathbf{E}\neg\varphi$
$\neg\mathbf{X}\varphi := \mathbf{X}\neg\varphi$	
$\neg(\varphi \mathbf{U} \psi) := \neg\varphi \mathbf{R} \neg\psi$	$\neg(\varphi \mathbf{R} \psi) := \neg\varphi \mathbf{U} \neg\psi$

where $p \in \text{Prop}$. ◁

The following proposition states that the logical complementation of a formula is true whenever the original formula is false. Its proof is completely standard and we will omit the details.

Proposition 2.9. Let \mathbb{S} be a transition system. Let $s \in S$ and $\varphi \in \text{CTL}_\Sigma^*$, then $\mathbb{S}, s \Vdash \neg\varphi$ iff $\mathbb{S}, s \not\Vdash \varphi$. Let $\pi \in \Pi(\mathbb{S})$ and $\psi \in \text{CTL}_\Pi^*$, then $\mathbb{S}, \pi \Vdash \neg\psi$ iff $\mathbb{S}, \pi \not\Vdash \psi$.

Proof. By analysis of the relational semantics. ◻

2.2 Board games

Throughout this paper mathematical games will be extensively used. The games featured in this paper are all two-player games played on colored game boards, which are (possibly infinite) graphs of which the vertices are colored with a finite set of colors. The two players are traditionally called Éloise and Abélard, and we will use the symbols \exists and \forall to represent them. In broad terms, it is the goal of \exists to show that something is true, whereas \forall tries to show that it is not. The use of these symbols \exists and \forall , which denote “there exists” and “for all” in first order logic, is deliberate; a victory for \exists often lies in her showing that there exists a solution and in those situations \forall wins if all attempted solutions fail. At the same time, the games in this paper are very symmetric, and we will make repeated use of this symmetry. For now, we focus on the abstract game theoretical notions that will play a role in this paper.

Definition 2.10. A *colored game board* is $(B_{\exists}, B_{\forall}, E, C, \Gamma)$ where $B := B_{\exists} \cup B_{\forall}$ is a set of *positions* divided between the players \exists and \forall , so that $B_{\exists} \cap B_{\forall} = \emptyset$, where $E \subseteq B \times B$ is a relation defining the moves, where C is a finite set of *colors* and where $\Gamma : B \rightarrow C$ is a coloring. A *match* is a finite or infinite sequence b_0, b_1, \dots such that $b_i E b_{i+1}$ for all applicable i . Every match *induces* a sequence c_0, c_1, \dots of colors by setting $c_i := \Gamma(b_i)$. For $b \in B$ we write $E[b] := \{b' \in B \mid b E b'\}$ for the set of *admissible moves* from b . \triangleleft

When defining the evaluation games and acceptance games in later sections, it will be convenient to have positions where there is only one admissible move. Since the move is necessary, we will not specify which of the two players must make the move.

For a finite match $m = b_0, \dots, b_n$ write $\text{last}(m) := b_n$. Now for every finite match m there is a player u such that $\text{last}(m) \in B_u$. Here u needs choose one of the admissible moves from $E[\text{last}(m)]$. If $E[\text{last}(m)]$ is empty then this is impossible, so match ends and u loses. If $E[\text{last}(m)]$ is not empty then the match must continue.

Definition 2.11. A match m is *complete* if m is infinite or if m is finite and $E[\text{last}(m)] = \emptyset$; it is *partial* otherwise. A complete finite match m is won by \exists if $\text{last}(m) \in B_{\forall}$ and won by \forall if $\text{last}(m) \in B_{\exists}$. \triangleleft

This gives us an elegant way to decide a winner for finite matches, but we still need to address infinite matches. A game therefore consists of a colored game board together with a winning condition that is based on colors. For an infinite match m let $\text{Inf}(m)$ denote the set of colors that occur infinitely often in the sequence induced by m . In this paper we consider two types of winning conditions and thus two types of games.

Definition 2.12. A *Muller game* is $\mathcal{G} = (B_{\exists}, B_{\forall}, E, C, \Gamma, \mathcal{F})$ where $(B_{\exists}, B_{\forall}, E, C, \Gamma)$ is a game board and where $\mathcal{F} \subseteq \wp(C)$ is a *Muller condition*. An infinite match m is won by \exists if $\text{Inf}(m) \in \mathcal{F}$ and won by \forall otherwise. \triangleleft

Definition 2.13. A *parity game* is $\mathcal{G} = (B_{\exists}, B_{\forall}, E, C, \Gamma, \Omega)$ where $(B_{\exists}, B_{\forall}, C, \Gamma, E)$ is a game board and where $\Omega : C \rightarrow \mathbb{N}$ is a *priority function*. An infinite match m is won by \exists if the maximum priority among colors in $\text{Inf}(m)$ is even and won by \forall if it is odd. \triangleleft

Note that this maximum is well-defined because C is finite. Also note that every parity game can be turned into a Muller game by having \mathcal{F} consist of those $X \subseteq C$ such that $\max\{\Omega(c) \mid c \in X\}$ is even. The converse is less evident, but in some cases a Muller game can also be expressed as a parity game. Parity games have some desirable properties that Muller games do not, so it will sometimes be useful to turn a Muller game into a parity game.

A key feature of this game theoretic framework is that we can discuss strategies. A strategy for a player u tells u what to do when it is their turn to move. Our two players have a perfect memory and there is no hidden information, so a strategy may use the entire history of the match. Let $\text{PM}_u \subseteq B^*$ denote the set of partial matches m of \mathcal{G} for which $\text{last}(m) \in B_u$.

Definition 2.14. A *strategy* of a player u for a game \mathcal{G} is a map $f : \text{PM}_u \rightarrow B$ such that $f(m) \in E[\text{last}(m)]$ for all $m \in \text{PM}_u$. A strategy f is *positional* if $f(m) = f(m')$ for all $m, m' \in \text{PM}_u$ with $\text{last}(m) = \text{last}(m')$. A match b_0, b_1, \dots is *consistent with f* if $b_{i+1} = f(b_0, \dots, b_i)$ for all i for which $b_i \in B_u$. A strategy f of u is *winning* if u wins every complete match that is consistent with f . \triangleleft

Strategies are hardly ever winning in this broad sense, and we will instead look at strategies which are winning when the starting position is fixed.

Definition 2.15. An *initialized game* is $\mathcal{G}@b$ where \mathcal{G} is a game and b is a board position. A match of $\mathcal{G}@b$ is a match b_0, b_1, \dots of \mathcal{G} such that $b_0 = b$. A winning strategy of a player u for $\mathcal{G}@b$ is a strategy f of u such that u wins every complete match of $\mathcal{G}@b$. A position b is *winning* for u if u has a winning strategy for $\mathcal{G}@b$. Let $\text{Win}_u(\mathcal{G}) \subseteq B$ denote the set of winning positions for u . \triangleleft

Clearly $\text{Win}_\exists(\mathcal{G}) \cap \text{Win}_\forall(\mathcal{G}) = \emptyset$ because a match of $\mathcal{G}@b$ cannot be won by both players at the same time. On the other hand it is not immediate that $\text{Win}_\exists(\mathcal{G}) \cup \text{Win}_\forall(\mathcal{G}) = B$, i.e. that for every position one of players has a winning strategy. In fact this only holds for games that enjoy determinacy.

Definition 2.16. A game \mathcal{G} enjoys *determinacy* if for every position $b \in B$ either \exists or \forall has a winning strategy for $\mathcal{G}@b$. A game enjoys *positional determinacy* if for every position $b \in B$ either \exists or \forall has a positional winning strategy for $\mathcal{G}@b$. \triangleleft

Determinacy is essential when using games to define semantics. This is because when the positions of a game represent logical statements, we will interpret $\text{Win}_\exists(\mathcal{G})$ as the set of statements that are true and $\text{Win}_\forall(\mathcal{G})$ as the set of statements that are false. Luckily the games discussed in this paper all enjoy determinacy.

Theorem 2.17. *All Muller games enjoy determinacy.*

Proof. By Zielonka (1998). \square

Theorem 2.18. *All parity games enjoy positional determinacy.*

Proof. By Emerson and Jutla (1991). \square

2.3 The modal μ -calculus

The modal μ -calculus is obtained from basic modal logic by adding variables and fixpoint operators. Throughout this paper we will assume Var is countably infinite set of variables, disjoint from Prop .

Definition 2.19. The syntax of $\mu\text{ML}(Y)$ is generated by the grammar

$$\varphi ::= \ell \mid y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \diamond\varphi \mid \square\varphi \mid \mu x.\vartheta \mid \nu x.\vartheta$$

where $\ell \in \text{Lit}$, $y \in Y$, $x \in \text{Var}$ and $\vartheta \in \mu\text{ML}(Y \cup \{x\})$. The syntax of μML is given by $\mu\text{ML}(\emptyset)$. \triangleleft

We will use λ to mean either fixpoint operator, thus $\lambda x.\vartheta$ refers to both $\mu x.\vartheta$ and $\nu x.\vartheta$. For every fixpoint formula $\lambda x.\vartheta$, we say that this formula *binds* x . Variables that are not bound by a fixpoint formula are free. The sets $\text{FV}(\varphi)$ and $\text{BV}(\varphi)$ of respectively free and bound variables in φ can be given by induction.

Definition 2.20. The *free* and *bound* variables of a $\mu\text{ML}(Y)$ formula are defined by

$$\begin{array}{ll} \text{FV}(\ell) := \emptyset & \text{BV}(\ell) := \emptyset \\ \text{FV}(y) := \{y\} & \text{BV}(y) := \emptyset \\ \text{FV}(\varphi \vee \psi) := \text{FV}(\varphi) \cup \text{FV}(\psi) & \text{BV}(\varphi \vee \psi) := \text{BV}(\varphi) \cup \text{BV}(\psi) \\ \text{FV}(\varphi \wedge \psi) := \text{FV}(\varphi) \cup \text{FV}(\psi) & \text{BV}(\varphi \wedge \psi) := \text{BV}(\varphi) \cup \text{BV}(\psi) \\ \text{FV}(\diamond\varphi) := \text{FV}(\varphi) & \text{BV}(\diamond\varphi) := \text{BV}(\varphi) \\ \text{FV}(\square\varphi) := \text{FV}(\varphi) & \text{BV}(\square\varphi) := \text{BV}(\varphi) \\ \text{FV}(\lambda x.\varphi) := \text{FV}(\varphi) \setminus \{x\} & \text{BV}(\lambda x.\varphi) := \text{BV}(\varphi) \cup \{x\} \end{array}$$

where $\ell \in \text{Lit}$ and $y \in Y$. \triangleleft

Although formally allowed in μML , formulas where a single variable is bound by multiple fixpoint operators are a bit hard to work with. Through this paper we will assume that $\text{FV}(\varphi)$ and $\text{BV}(\varphi)$ are disjoint and that every bound variable is only bound by a single subformula. This creates a natural order on the bound variables by the complexity of their binding formula.

Definition 2.21. A $\mu\text{ML}(Y)$ formula is *clean* if no two fixpoint subformulas bind the same variable and no free variable is bound by a fixpoint subformula. If ξ is a clean μML formula then for every bound variable x there is a unique fixpoint formula $\lambda_\xi(x) := \lambda_x x.\vartheta_x$ that binds it. Define the relation \prec_ξ on $\text{BV}(\xi)$ by having $x \prec_\xi y$ whenever $\lambda_\xi(x)$ is a proper subformula of $\lambda_\xi(y)$. Write $x \preceq_\xi y$ whenever $x \prec_\xi y$ or $x = y$. \triangleleft

Note that \prec_ξ is a transitive irreflexive relation and that \preceq_ξ is a transitive reflexive antisymmetric relation.

The semantics for μML as defined by Kozen (1983) are algebraic, i.e. based on a map that assigns to each formula φ its meaning $\llbracket \varphi \rrbracket^{\mathbb{S}}$ on a transition system \mathbb{S} . In Emerson and Jutla (1991) these algebraic semantics were shown to be equivalent to semantics based on evaluation games. Evaluation game semantics proved to be much more convenient when comparing μML with automata, as we will do in this paper. Therefore we will forgo the algebraic semantics entirely and focus on evaluation game semantics.

Evaluation games are games played on a board $S \times \Phi$, where S is the set of states of a transition system \mathbb{S} and where Φ is a finite set of formulas. A position (s, φ) is intended to be winning for \exists whenever φ is true at s . The set Φ also serves as the set of colors, with a natural coloring that sends (s, φ) to φ . A match of an evaluation game therefore induces a sequence of formulas from Φ . We will call such sequences *traces*. For a μML formula ξ , the formulas that occur in a trace will be subformulas of ξ .

Definition 2.22. For a μML formula ξ , let $\text{Sb}(\xi)$ denote its set of subformulas. \triangleleft

Traces will generally start at some φ_0 , then go to some subformula φ_1 of φ_0 , then to some subformula φ_2 of φ_1 , etcetera. When a bound variable x is reached, the trace will continue to its binding formula. Any other formula will be followed by one of its direct proper subformulas.

Definition 2.23. Let ξ be a μML formula and let $\varphi \in \text{Sb}(\xi)$. Define the set $\nabla_{\xi}(\varphi)$ of ξ -derivatives of φ as follows:

$$\begin{aligned} \nabla_{\xi}(\ell) &:= \emptyset \\ \nabla_{\xi}(x) &:= \{\lambda_{\xi}(x)\} \\ \nabla_{\xi}(\varphi \star \psi) &:= \{\varphi, \psi\} && \text{where } \star \text{ is one of } \vee, \wedge \\ \nabla_{\xi}(\tau\varphi) &:= \{\varphi\} && \text{where } \tau \text{ is one of } \diamond, \square \\ \nabla_{\xi}(\lambda x.\varphi) &:= \{\varphi\} && \text{where } \lambda \text{ is one of } \mu, \nu \end{aligned}$$

where $\ell \in \text{Lit}$ and $x \in \text{BV}(\xi)$. For $\varphi, \psi \in \text{Sb}(\xi)$, write $\varphi \triangleleft_{\xi} \psi$ whenever there is a sequence $\varphi_0, \dots, \varphi_n$ with $\varphi_0 = \varphi$, $\varphi_n = \psi$ and $\varphi_i \in \nabla_{\xi}(\varphi_{i+1})$. Write $\varphi \bowtie_{\xi} \psi$ when $\varphi \triangleleft_{\xi} \psi$ and $\psi \triangleleft_{\xi} \varphi$. \triangleleft

Note that \triangleleft_{ξ} is a transitive reflexive relation and that \bowtie_{ξ} is an equivalence relation. Also note that $\text{Sb}(\xi)$ is closed under ξ -derivatives, i.e. $\nabla_{\xi}(\varphi) \subseteq \text{Sb}(\xi)$ for all $\varphi \in \text{Sb}(\xi)$, and therefore under \triangleleft_{ξ} and \bowtie_{ξ} .

Definition 2.24. Let ξ be a μML formula. A ξ -trace is a finite or infinite sequence $\varphi_0, \varphi_1, \dots$ of formulas such that $\varphi_i \bowtie_{\xi} \varphi_{i+1}$ for all i . An infinite trace is *stalling* if there is k such that $\varphi_i = \varphi_{i+1}$ for all $i \geq k$. A trace is *complete* if it either ends with a literal or free variable or is non-stalling. A trace $\varphi_0, \varphi_1, \dots$ is *direct* if $\varphi_{i+1} \in \nabla_{\xi}(\varphi_i)$ for all i . \triangleleft

Note that every direct trace is complete. As one might expect, complete matches will induce complete traces. For traces that end in a literal, the winner can be determined by whether that literal is true or not. For infinite traces the winner will depend on the bound variable with the most complex binding formula.

position	player	admissible moves	
(s, ℓ)	\forall	\emptyset	if $s \Vdash_V \ell$
(s, ℓ)	\exists	\emptyset	if $s \not\Vdash_V \ell$
(s, x)	-	$\{(s, \lambda_x(x))\}$	
$(s, \varphi_1 \vee \varphi_2)$	\exists	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \varphi_1 \wedge \varphi_2)$	\forall	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \diamond\varphi)$	\exists	$\{(t, \varphi) \mid sRt\}$	
$(s, \square\varphi)$	\forall	$\{(t, \varphi) \mid sRt\}$	
$(s, \mu x.\varphi)$	-	$\{(s, \varphi)\}$	
$(s, \nu x.\varphi)$	-	$\{(s, \varphi)\}$	

Figure 2.1: The rules for the evaluation game of μML formulas.

Proposition 2.25. *Let ξ be a μML formula. For every infinite direct ξ -trace $\varphi_0, \varphi_1, \dots$ there is a \preceq_ξ -greatest bound variable of ξ that occurs infinitely often.*

Proof. By Emerson and Jutla (1991). □

We are now ready to define the evaluation game for μML .

Definition 2.26. Let ξ be a μML formula and let $\mathbb{S} = (S, R, V)$ be a transition system. The *evaluation game for ξ on \mathbb{S}* , denoted $\mathcal{E}(\xi, \mathbb{S})$, is played on the board

$$S \times \text{Sb}(\xi)$$

according to the rules given by Figure 2.1. Every match induces a direct ξ -trace. Now Proposition 2.25 states that there is a \preceq_ξ -greatest bound variable x that occurs infinitely often during the match. The winner is decided by the type of fixpoint that bound x ; \exists wins if $\lambda_x = \nu$ and \forall wins if $\lambda_x = \mu$. ◁

We define the meaning of a formula φ on a transition system \mathbb{S} as the set of states $s \in S$ for which (s, φ) is a winning position for \exists in the game $\mathcal{E}(\varphi, \mathbb{S})$.

Definition 2.27. Let \mathbb{S} be a transition system. Define satisfaction of a formula $\varphi \in \mu\text{ML}$ at a state $s \in S$ by $\mathbb{S}, s \Vdash \varphi$ iff $(s, \varphi) \in \text{Win}_\exists(\mathcal{E}(\varphi, \mathbb{S}))$. ◁

With this in mind, a roles of \exists and \forall as verifier and falsifier respectively shine through the rules in Figure 2.1. For instance if \exists wants to show that a formula $\diamond\varphi$ holds at s , then she needs to be able to find a state t reachable from s such that φ holds at t . Conversely if $\square\varphi$ is to hold at s , then φ needs to hold at every successor t of s , and it is the task of \forall to choose such a t . This also illustrates the duality of \exists and \forall : if \forall wants to show that $\square\varphi$ does not hold at s , then he needs to find a successor t of s where φ does not hold. Therefore it would be nice if $\mathbb{S}, s \not\Vdash \varphi$ iff $(s, \varphi) \in \text{Win}_\forall(\mathcal{E}(\varphi, \mathbb{S}))$. This in fact follows from the determinacy of evaluation games, which can be given by defining a priority based on fixpoint depth.

Definition 2.28. Define the *fixpoint-depth* of a $\mu\text{ML}(Y)$ formula, $\text{fd}(\varphi)$, by

$$\begin{aligned} \text{fd}(\ell) &:= 0 \\ \text{fd}(y) &:= 0 \\ \text{fd}(\varphi \star \psi) &:= \max\{\text{fd}(\varphi), \text{fd}(\psi)\} && \text{where } \star \text{ is one of } \vee, \wedge \\ \text{fd}(\tau\varphi) &:= \text{fd}(\varphi) && \text{where } \tau \text{ is one of } \diamond, \square \\ \text{fd}(\lambda x.\varphi) &:= \text{fd}(\varphi) + 1 && \text{where } \lambda \text{ is one of } \mu, \nu \end{aligned}$$

where $\ell \in \text{Lit}$ and $y \in Y$. ◁

Proposition 2.29. *Evaluation games for μML formulas enjoy positional determinacy.*

Proof. Let ξ be a μML formula. Define a priority Ω_ξ on $\text{Sb}(\xi)$ by

$$\begin{aligned} \Omega_\xi(x) &:= 2 \cdot \text{fd}(\vartheta_x) + 1 && \text{if } \lambda_x = \mu \\ \Omega_\xi(x) &:= 2 \cdot \text{fd}(\vartheta_x) + 2 && \text{if } \lambda_x = \nu \end{aligned}$$

for $x \in \text{BV}(\xi)$ and $\Omega_\xi(\varphi) := 0$ for other formulas.

Note that if $x \prec_\xi y$ then $\Omega_\xi(x) < \Omega_\xi(y)$. This means that if x is the \preceq_ξ -greatest element of a set of bound variables, then its priority is the highest priority among variables from that set. Furthermore note that $\Omega_\xi(x)$ is odd if $\lambda_x = \mu$ and even if $\lambda_x = \nu$. The winning condition from Definition 2.26 is therefore equivalent to a parity winning condition based on Ω_ξ . The result now follows from Theorem 2.18. ◻

To fully appreciate the duality between \exists and \forall , we can look at logical complementation.

Definition 2.30. Define the *logical complement* $\neg\varphi$ of a $\mu\text{ML}(Y)$ formula φ by

$$\begin{aligned} \neg\top &:= \perp && \neg\perp &:= \top \\ \neg(p) &:= \neg p && \neg(\neg p) &:= p \\ \neg y &:= y \\ \neg(\varphi \vee \psi) &:= \neg\varphi \wedge \neg\psi && \neg(\varphi \wedge \psi) &:= \neg\varphi \vee \neg\psi \\ \neg\diamond\varphi &:= \square\neg\varphi && \neg\square\varphi &:= \diamond\neg\varphi \\ \neg(\mu x.\vartheta) &:= \nu x.\neg\vartheta && \neg(\nu x.\vartheta) &:= \mu x.\neg\vartheta \end{aligned}$$

where $p \in \text{Prop}$ and $y \in Y$. ◁

The complementation rule for variables might seem unintuitive at first. However, if we interpret fixpoint operators as binary operators between literals and formulas, then the complement of a formula such as $\varphi := \mu x.(p \wedge \diamond x)$ would be $\nu(\neg x).(\neg p \vee \square(\neg x))$. If we treat $\neg x$ as another variable, say y , then this is just $\nu y.(\neg p \vee \square y)$. This is in fact equivalent to the real complement of φ , which is $\nu x.(\neg p \vee \square x)$.

Now the following proposition states that logical complementation is indeed the same as switching the roles of \exists and \forall . We will omit the details of its proof.

Proposition 2.31. *Let \mathbb{S} be a transition system. Let $s \in S$ and $\varphi \in \mu\text{ML}$, then $\mathbb{S}, s \Vdash \neg\varphi$ iff $\mathbb{S}, s \not\Vdash \varphi$.*

Proof. By Emerson and Jutla (1991). ◻

2.4 Stream automata

Although we will mostly work with modal automata in this paper, let us first look at stream automata. Stream automata work not just on Σ -words, i.e. finite sequences of symbols from some finite alphabet Σ , but particularly on infinite Σ -streams.

Definition 2.32. A *parity Σ -stream automaton* is $\mathbb{B} = (B, \delta, \Omega, b_{\text{I}})$ where B is a finite set of states, where $\delta \subseteq B \times \Sigma \times B$ is a transition relation, where $\Omega : B \rightarrow \mathbb{N}$ is a priority function and where $b_{\text{I}} \in B$ is the initial state. \triangleleft

The set B together with its transition relation δ can be seen as a directed graph whose edges are labelled with symbols from Σ . A finite walk through this graph then becomes labelled by some word over Σ .

Definition 2.33. Let $\mathbb{B} = (B, \delta, \Omega, b_{\text{I}})$ be an Σ -stream automaton and let $b, b' \in B$. For a symbol $\sigma \in \Sigma$, write $b \rightarrow_{\sigma} b'$ whenever $(b, \sigma, b') \in \delta$. For an Σ -word $w = \sigma_1, \dots, \sigma_n$, write $b \rightarrow_w b'$ when there are b_0, \dots, b_n with $b_0 = b$, $b_n = b'$ and $b_i \rightarrow_{\sigma_i} b_{i+1}$ for each $i < n$. \triangleleft

On the other hand, for a given word it remains to be seen if there are finite walks through the graph that are labelled with that word; if there are, the automaton accepts that word. To find these walks, we start with the initial state and then read off one symbol at a time, taking one of the outgoing arrows labelled with that symbol. Generalizing this to an infinite stream of symbols, we get infinite runs.

Definition 2.34. A *run* of an automaton $\mathbb{B} = (B, \delta, \Omega, b_{\text{I}})$ on a stream $\sigma_0, \sigma_1, \dots$ is an infinite sequence b_0, b_1, \dots of states such that $b_0 = b_{\text{I}}$ and $b_i \rightarrow_{\sigma_i} b_{i+1}$ for all i . A run is *accepting* if the highest priority among states occurring infinitely often in the run is even. An automaton \mathbb{B} *accepts* a stream if there exists an accepting run of \mathbb{B} on the stream. \triangleleft

The above defined transition relation can be non-deterministic; for any state b and symbol σ , there may be multiple outgoing arrows from b labelled with σ , or none at all. When appropriate, deterministic stream automata can be used instead.

Definition 2.35. A stream automaton $\mathbb{B} = (B, \delta, \Omega, b_{\text{I}})$ is *deterministic* if δ is a total function, i.e. if for every $b \in B$ and $\sigma \in \Sigma$ there is exactly one $b' \in B$ with $b \rightarrow_{\sigma} b'$. \triangleleft

When defining deterministic stream automata we will write $\delta(b, \sigma) = b'$ instead of $(b, \sigma, b') \in \delta$. Note that for deterministic automata, there is exactly one run for every stream; a deterministic automaton accepts a stream if and only if this run is accepting.

2.5 Modal automata

Instead of on words or streams over some alphabet, modal automata work on transition systems. For modal automata, the role of the arrows between states that are labelled with symbols is fulfilled by a single modal ‘transition term’ for each state. These transition

terms are modal formulas that contain occurrences of states of the automaton. If a state b occurs in the transition term for a , this represents an arrow from a to b . To be more precise, the transition terms are *modal one-step formulas* over the set of states.

Definition 2.36. The syntax of $\text{ML}_1(A)$ is generated by the grammar

$$\varphi ::= \ell \mid a \mid \diamond a \mid \square a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

where $\ell \in \text{Lit}$ and $a \in A$. Occurrences of $a \in A$ of the form a , $\diamond a$ and $\square a$ are called *unguarded*, *\diamond -guarded* and *\square -guarded* respectively. A modal one-step formula is *guarded* and hence in $\text{gML}_1(A)$ if it contains no unguarded occurrences of $a \in A$. \triangleleft

In some papers modal automata are implicitly guarded and thus have transition terms from gML_1 ; in that context modal automata with transition terms from ML_1 are called ‘silent-step’ automata. However here it is more convenient to work with silent-step automata by default, and we will be explicit whenever we use guarded automata.

Definition 2.37. A *modal automaton* is $\mathbb{A} = (A, \Delta, \text{Acc}, a_1)$ where A is a finite set of states, where $\Delta : A \rightarrow \text{ML}_1(A)$ is a transition function, where Acc is some acceptance condition where $a_1 \in A$ is the initial state. \triangleleft

When $\mathbb{A} = (A, \Delta, \text{Acc}, a_1)$ is an automaton and $b \in A$ is a state, we will write (\mathbb{A}, b) to refer to the automaton $(A, \Delta, \text{Acc}, b)$ where the starting state has been replaced with b . The most general acceptance condition we will use in this paper is the Muller condition.

Definition 2.38. A *modal Muller automaton* is a modal automaton $\mathbb{A} = (A, \Delta, \mathcal{F}, a_1)$ where the acceptance condition is given by a Muller set $\mathcal{F} \subseteq \wp(A)$; an infinite sequence of states is accepted iff the set of states occurring infinitely often belongs to \mathcal{F} . \triangleleft

This corresponds in a natural way with the Muller games from section 2.2. Of particular interest are automata whose acceptance condition is a parity condition.

Definition 2.39. A *modal parity automaton* is a modal automaton $\mathbb{A} = (A, \Delta, \Omega, a_1)$ where the acceptance condition is given by a priority function $\Omega : A \rightarrow \mathbb{N}$; an infinite sequence of states is accepted iff the highest priority among states occurring infinitely often is even. \triangleleft

For modal automata, the definition of acceptance can best be given by game semantics.

Definition 2.40. Let \mathbb{A} be a modal automaton and let \mathbb{S} be a transition system. The *acceptance game for \mathbb{A} on \mathbb{S}* , denoted $\mathcal{A}(\mathbb{A}, \mathbb{S})$, is played on the board

$$S \times \text{ML}_1(A)$$

according to the rules given in Figure 2.2. Every infinite match induces an infinite sequence of states from A by looking at the positions of the form (t, a) . An infinite match is won by \exists iff the acceptance condition of \mathbb{A} accepts this sequence. \triangleleft

position	player	admissible moves	
(s, a)	-	$\{(s, \Delta(a))\}$	
(s, ℓ)	\forall	\emptyset	if $s \Vdash_V \ell$
(s, ℓ)	\exists	\emptyset	if $s \not\Vdash_V \ell$
$(s, \varphi_1 \vee \varphi_2)$	\exists	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \varphi_1 \wedge \varphi_2)$	\forall	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \diamond a)$	\exists	$\{(t, a) \mid sRt\}$	
$(s, \square a)$	\forall	$\{(t, a) \mid sRt\}$	

Figure 2.2: The rules for the acceptance game of modal automata.

It may be clear that the acceptance game of a Muller automaton is a Muller game, where the set of states A serves as the finite set of colors. Similarly, the acceptance game of a parity automaton is a parity game. We now define acceptance in terms of the acceptance game as initialized by a state of the transition system and the starting state of the automaton.

Definition 2.41. Let \mathbb{A} be a modal automaton and let \mathbb{S} be a transition system. The automaton \mathbb{A} *accepts* a state $s \in S$, denoted $\mathbb{S}, s \Vdash \mathbb{A}$, whenever $(s, a_1) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$. \triangleleft

The reversed notation of $\mathbb{S}, s \Vdash \mathbb{A}$ is appropriate in this context because automata – like formulas – are used to encode properties of transition systems. As with formulas, we will write $\mathbb{A} \equiv \mathbb{A}'$ when two automata accept the same states, i.e. when $\mathbb{S}, s \Vdash \mathbb{A}$ iff $\mathbb{S}, s \Vdash \mathbb{A}'$ for all transition systems \mathbb{S} and all $s \in S$. With some abuse of notation, we will also write $\mathbb{A} \equiv \varphi$ when an automaton is logically equivalent to a (state) formula. Again it seems intuitive to have that $\mathbb{S}, s \not\Vdash \mathbb{A}$ whenever $(s, a_1) \in \text{Win}_{\forall}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$; this holds because the acceptance games of automata enjoy determinacy.

Proposition 2.42. *Acceptance games for modal Muller automata enjoy determinacy.*

Proof. This follows from Theorem 2.17. \square

Proposition 2.43. *Acceptance games for modal parity automata enjoy positional determinacy.*

Proof. This follows from Theorem 2.18. \square

Although the transitions between states are given by formulas instead of relations, it is still useful to consider concepts such as reachability.

Definition 2.44. Let \mathbb{A} be an automaton. Define \rightarrow on A by $a \rightarrow b$ whenever b occurs in $\Delta(a)$. Define \rightarrow^* to be the transitive reflexive closure of \rightarrow . Define \rightarrow^* on A by $a \rightarrow^* b$ whenever $a \rightarrow^* b$ and $b \not\rightarrow^* a$. \triangleleft

Note that \rightarrow^* is transitive, irreflexive and hence asymmetric. The \rightarrow^* relation indicates that a state can ‘descend’ to another state, after which point it cannot go back. If two states can reach each other, then they are part of the same ‘cluster’.

Definition 2.45. Let \mathbb{A} be a modal automaton. A *cluster* of \mathbb{A} is a maximal subset $C \subseteq A$ such that $a \rightarrow b$ for all $a, b \in C$. A cluster C is *trivial* if $C = \{c\}$ and $c \not\rightarrow c$. \triangleleft

Since A is finite and \rightarrow is both transitive and asymmetric, \rightarrow is converse-wellfounded, i.e. there is no infinite sequence $a_0 \rightarrow a_1 \rightarrow \dots$ in \mathbb{A} . This induces a natural notion of depth on clusters and on the states within them, which decreases whenever a match of the acceptance game descends to a lower cluster. The ‘cluster-depth’ of a state is the maximum number of \rightarrow -steps that can be taken, plus one if the resulting cluster is non-trivial.

Definition 2.46. The *cluster-depth* of a state $a_0 \in A$ is the length of the longest sequence $a_1, \dots, a_n \in A$ such that $a_i \rightarrow a_{i+1}$ for all $i < n - 1$ and $a_{n-1} \rightarrow a_n$. \triangleleft

Note that the cluster-depth of any two elements of a cluster is the same, and that if $a \rightarrow b$ then the cluster-depth of b is strictly less than that of a . This allows us to perform induction on the cluster-depth of states. For the base case of such an induction, note that only trivial clusters have elements with cluster-depth 0. Other than clusters, we will sometimes need a slight generalization of clusters.

Definition 2.47. Let \mathbb{A} be a modal automaton. A *generalized cluster* of \mathbb{A} is a subset $X \subseteq A$ such that if $a \in X$ and $a \rightarrow b$ then either $b \in X$ or $b \not\rightarrow a$. \triangleleft

Note that every cluster is a generalized cluster. The converse does not hold; in particular an automaton \mathbb{A} might have many different clusters but the entire set A will always be a generalized cluster. Unlike clusters, generalized clusters might overlap and might be contained in one another.

The number, size and depth of clusters all attribute to the complexity and thus power of an automaton. The complexity of a parity automaton can also be influenced by its Mostowski index, which is the range of priorities that the priority map can use. The smaller this range, the less powerful the automaton. We also define Mostowski indices for individual clusters of an automaton.

Definition 2.48. The *Mostowski index* of a modal parity automaton \mathbb{A} is (l, h) where l is the lowest priority among states of \mathbb{A} and h the highest. The Mostowski index of a cluster C of \mathbb{A} is (l, h) where l is the lowest priority among states in C and h the highest. \triangleleft

For given $x, y \in \mathbb{N}$, we will say that an automaton has a Mostowski index inside (x, y) if its Mostowski index is (l, h) for some $l, h \in \mathbb{N}$ with $x \leq l \leq h \leq y$. Similarly we say that an automaton has a Mostowski index of at most y if its index is (l, h) where $h \leq y$. Of particular interest are the Mostowski indices $(1, 2)$ and $(0, 1)$.

Definition 2.49. A cluster of a modal parity automaton is *Büchi* if it has a Mostowski index inside $(1, 2)$. It is *co-Büchi* if it has a Mostowski index inside $(0, 1)$. \triangleleft

Lastly we note that priorities are relative. That is, we are free to increase or decrease all priorities by some natural number, as long as the parities and the relative order of the priorities remain the same.

Proposition 2.50. *Let $\mathbb{A} = (A, \Delta, \Omega, a_1)$ be a modal parity automaton. Let $\Omega' : A \rightarrow \mathbb{N}$ be a priority map such that if $\Omega(a) < \Omega(b)$ then $\Omega'(a) < \Omega'(b)$ and such that the parity of $\Omega'(a)$ is the parity of $\Omega(a)$. Then $\mathbb{A} \equiv (A, \Delta, \Omega', a_1)$.*

Proof. Let \mathbb{A} be a modal parity automaton and let Ω' be such a priority map. Define $\mathbb{A}' := (A, \Delta, \Omega', a_1)$. Let \mathbb{S} be a transition system. Clearly any match of $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is also a match of $\mathcal{A}(\mathbb{A}', \mathbb{S})$ and vice versa, so it remains to check that the winner is the same in both games. Let m be an infinite match, then there are states a and b that occur infinitely often during the match such that $\Omega(a) \geq \Omega(c)$ and $\Omega'(b) \geq \Omega'(c)$ for all c that occur infinitely often. In particular $\Omega'(b) \geq \Omega'(a)$, hence $\Omega'(b) \not< \Omega'(a)$ and thus $\Omega(b) \not< \Omega(a)$. But also $\Omega(b) \leq \Omega(a)$, so it must be that $\Omega(b) = \Omega(a)$. Now the parity of $\Omega'(b)$ must be equal to the parity of $\Omega(b) = \Omega(a)$. This means that m is won by the same player in both games. \square

Chapter 3

Dominance in evaluation games

At first glance the logics CTL^* and μML may seem very different; one operates both on states and on paths, whereas the other uses variables. However the two logics have in common that their central operators, U and R for CTL^* and μ and ν for μML , reveal an unfolding behavior. For CTL^* this behavior is exemplified by the equivalences

$$\begin{aligned}\alpha\text{U}\beta &\equiv \beta \vee (\alpha \wedge \text{X}(\alpha\text{U}\beta)) \\ \alpha\text{R}\beta &\equiv \beta \wedge (\alpha \vee \text{X}(\alpha\text{R}\beta))\end{aligned}$$

for all $\alpha, \beta \in \text{CTL}^*$. For μML , it is given by the equivalence

$$\lambda x.\vartheta \equiv \vartheta[\lambda x.\vartheta/x]$$

for every fixpoint formula $\lambda x.\vartheta \in \mu\text{ML}(\emptyset)$, where the right-hand formula is created from $\vartheta \in \mu\text{ML}(\{x\})$ by replacing every free occurrence of x with $\lambda x.\vartheta$. As a specific example, for $\varphi := \mu x.(p \vee \diamond x)$ the equivalence is $\varphi \equiv p \vee \diamond \varphi$.

Note that the occurrence of φ in its unfolding is guarded by the modality \diamond because x was \diamond -guarded. Likewise, the occurrences of U and R in their unfoldings can be seen as being guarded by the operator X . In this way unfolding allows us to break down a semantically complicated formulas into a “now” and a “later”. This will play a key role in the construction of automata for CTL^* .

Lastly, observe that unfolding for both U and μ is in a sense finite, whereas for R and ν it is infinite. This suggests that it is possible to encode the formula $p\text{U}q$ in μML as $\mu x.(q \vee (p \wedge \text{X}x))$ and dually $p\text{R}q$ as $\nu x.(q \wedge (p \vee \text{X}x))$, except that the operator X is not part of μML . Indeed much of the complexity of the translation of CTL^* into μML by Dam (1990) and the translation from CTL^* into automata in this paper, lies in the apparent inability to evaluate modal formulas on paths.

3.1 Evaluation games for CTL^*

In order to better integrate CTL^* with the other frameworks in this paper, we will define evaluation game semantics for CTL^* as well. As with μML , the positions of the evaluation

game for CTL* will be of the form (s, φ) where s is a state in a transition system and φ a formula from a finite set Φ . Additionally, there are positions (π, φ) where π is a path. The matches of the evaluation match will form direct traces; that is, each formula will be followed by one of its derivatives.

Definition 3.1. Define the set $\nabla\varphi$ of *derivatives* of a CTL* formula φ as follows:

$$\begin{aligned}
\nabla(\ell) &:= \emptyset \\
\nabla(\varphi \star \psi) &:= \{\varphi, \psi\} && \text{where } \star \text{ is one of } \vee, \wedge \\
\nabla(Q\varphi) &:= \{\varphi\} && \text{where } Q \text{ is one of } E, A \\
\nabla(X\varphi) &:= \{\varphi\} \\
\nabla(\varphi U\psi) &:= \{\psi \vee (\varphi \wedge X(\varphi U\psi))\} \\
\nabla(\varphi R\psi) &:= \{\psi \wedge (\varphi \vee X(\varphi R\psi))\}
\end{aligned}$$

where $\ell \in \text{Lit}$. ◁

Note that as CTL* does not have variables and bindings, the derivatives no longer depend on a starting formula. On the other hand, if we start with a U-formula then its derivative is not a subformula. Instead of taking $\text{Sb}(\xi)$ the finite set of states, we will take the (*Fischer-Ladner*) closure of ξ .

Definition 3.2. The *closure* $\text{Cl}(\Phi)$ of a set of CTL* formulas Φ is the least set of formulas that contains Φ and is closed under derivatives. ◁

We write $\text{Cl}_\Sigma(\Phi) := \text{Cl}(\Phi) \cap \text{CTL}_\Sigma^*$, $\text{Cl}_\Pi(\Phi) := \text{Cl}(\Phi) \cap \text{CTL}_\Pi^*$ and $\text{Cl}(\varphi) := \text{Cl}(\{\varphi\})$. Naturally every derivative of a CTL* formula is itself a CTL* formula; in this sense CTL* is closed under taking derivatives. However CTL is not closed; for example pUq is a CTL_Π formula, but its derivative $q \vee (p \wedge X(pUq))$ is not in CTL. On the other hand, $\text{Cl}(\text{CTL})$ is still much smaller than CTL*, as we will see at the end of this section. First, we will need that $\text{Cl}(\xi)$ is finite for every CTL* formula ξ .

Proposition 3.3. *If $\Phi \subseteq \text{CTL}^*$ is finite then $\text{Cl}(\Phi)$ is finite.*

Proof. First observe that Cl is a finitary closure operator, such that $\text{Cl}(\Phi) = \bigcup_{\varphi \in \Phi} \text{Cl}(\varphi)$. In particular $\text{Cl}(\varphi) = \{\varphi\} \cup \bigcup_{\psi \in \nabla\varphi} \text{Cl}(\psi)$. An induction on the structure of φ then shows that $\text{Cl}(\varphi)$ is finite for every φ . We highlight the case where $\varphi = \alpha U\beta$. Here

$$\begin{aligned}
\text{Cl}(\alpha U\beta) &= \{\alpha U\beta\} \cup \text{Cl}(\beta \vee (\alpha \wedge X(\alpha U\beta))) \\
&= \{\alpha U\beta, \beta \vee (\alpha \wedge X(\alpha U\beta))\} \cup \text{Cl}(\alpha \wedge X(\alpha U\beta)) \cup \text{Cl}(\beta) \\
&= \{\alpha U\beta, \alpha \wedge X(\alpha U\beta), \beta \vee (\alpha \wedge X(\alpha U\beta))\} \cup \text{Cl}(X(\alpha U\beta)) \cup \text{Cl}(\alpha) \cup \text{Cl}(\beta) \\
&= \{\alpha U\beta, X(\alpha U\beta), \alpha \wedge X(\alpha U\beta), \beta \vee (\alpha \wedge X(\alpha U\beta))\} \cup \text{Cl}(\alpha) \cup \text{Cl}(\beta)
\end{aligned}$$

which is finite if $\text{Cl}(\alpha)$ and $\text{Cl}(\beta)$ are finite. ◻

As derivatives do not depend on the starting formula, we can define the relations \triangleleft and \bowtie on the entirety of CTL^* . Write $\varphi \triangleleft \psi$ when $\varphi \in \text{Cl}(\psi)$, and $\varphi \bowtie \psi$ when $\varphi \triangleleft \psi$ and $\varphi \triangleright \psi$. Note that \triangleleft is a transitive reflexive relation and that \bowtie is an equivalence relation. As ξ -traces were sequences that follow \triangleright_ξ , so are traces of CTL^* sequences that follow \triangleright .

Definition 3.4. A *trace* through CTL^* is a finite or infinite sequence $\varphi_0, \varphi_1, \dots$ of formulas such that $\varphi_i \triangleright \varphi_{i+1}$ for all i . An infinite trace is *stalling* if there is k such that $\varphi_i = \varphi_{i+1}$ for all $i \geq k$. A trace is *complete* if it either ends with a literal or is non-stalling. A trace $\varphi_0, \varphi_1, \dots$ is *direct* if $\varphi_{i+1} \in \nabla\varphi_i$ for all i . \triangleleft

Before we can define the evaluation games, we will need to address infinite traces. First observe that every trace eventually ends up in an equivalence class of \bowtie .

Proposition 3.5. *For every trace $\varphi_0, \varphi_1, \dots$ there is k such that $\varphi_i \bowtie \varphi_j$ for all $i, j \geq k$.*

Proof. Let $\varphi_0, \varphi_1, \dots$ be an infinite trace. Clearly $\varphi_i \in \text{Cl}(\varphi_0)$ for all i , which is finite by Proposition 3.3. Thus there must be a $\psi \in \text{Cl}(\varphi_0)$ that occurs infinitely often along the trace. Let k be the index of its first occurrence, then for every $i \geq k$ there is $j \geq i$ with $\varphi_j = \psi$; now $\psi = \varphi_k \triangleright \varphi_i \triangleright \varphi_j = \psi$ hence $\varphi_i \bowtie \psi$. This gives $\varphi_i \bowtie \psi \bowtie \varphi_j$ for all $i, j \geq k$. For a finite trace $\varphi_0, \dots, \varphi_n$, take $k = n$. \square

In order to determine a winner for infinite traces, we thus need to classify the equivalence classes of \bowtie as “good” or “bad” for \exists . In fact, it turns out that there are only two kinds to non-trivial classes.

Proposition 3.6. *The non-trivial equivalence classes of \bowtie on CTL^* are of the form*

$$\{\alpha\text{U}\beta, \text{X}(\alpha\text{U}\beta), \alpha \wedge \text{X}(\alpha\text{U}\beta), \beta \vee (\alpha \wedge \text{X}(\alpha\text{U}\beta))\}$$

or

$$\{\alpha\text{R}\beta, \text{X}(\alpha\text{R}\beta), \alpha \vee \text{X}(\alpha\text{R}\beta), \beta \wedge (\alpha \vee \text{X}(\alpha\text{R}\beta))\}$$

for some α, β .

Proof. Note that the derivatives of a CTL^* formula φ are proper subformulas of φ , unless φ is a U -formula or an R -formula. If $\varphi \bowtie \psi$ and $\varphi \neq \psi$ then there are $\chi_0, \dots, \chi_n, \dots, \chi_{n+k}$ such that $\chi_0 = \varphi$, $\chi_n = \psi$, $\chi_{n+k} = \varphi$ and $\chi_i \in \nabla\chi_{i+1}$ for all $i < n+k$. Note that $\varphi \bowtie \chi_i \bowtie \psi$ for all $i \leq n+k$. It cannot be that each χ_i is a proper subformula of χ_{i+1} , because then φ would have to be a proper subformula of itself. Therefore there must be some i for which χ_i is either a U -formula or an R -formula. We conclude that any non-trivial equivalence class of \bowtie contains such a formula.

Define the UR -depth of a CTL^* formula, $\text{urd}(\varphi)$, by

$$\begin{aligned} \text{urd}(\ell) &:= 0 \\ \text{urd}(\varphi \star \psi) &:= \max\{\text{urd}(\varphi), \text{urd}(\psi)\} & (\star \in \{\vee, \wedge\}) \\ \text{urd}(\tau\varphi) &:= \text{urd}(\varphi) & (\tau \in \{\text{E}, \text{A}, \text{X}\}) \\ \text{urd}(\varphi \star \psi) &:= \max\{\text{urd}(\varphi), \text{urd}(\psi)\} + 1 & (\star \in \{\text{U}, \text{R}\}) \end{aligned}$$

and note that $\text{urd}(\varphi) \leq \text{urd}(\psi)$ whenever $\varphi \in \nabla\psi$, and therefore $\text{urd}(\varphi) \leq \text{urd}(\psi)$ whenever $\varphi \triangleleft \psi$. Let $\alpha\mathbf{U}\beta$ be a formula of CTL^* , then $\text{urd}(\varphi) \leq \text{urd}(\alpha) < \text{urd}(\alpha\mathbf{U}\beta)$ and hence $\text{urd}(\alpha\mathbf{U}\beta) \not\leq \text{urd}(\varphi)$ for all $\varphi \in \text{Cl}(\alpha)$, and similarly $\text{urd}(\alpha\mathbf{U}\beta) \not\leq \text{urd}(\varphi)$ for all $\varphi \in \text{Cl}(\beta)$. Thus if $\varphi \triangleright \alpha\mathbf{U}\beta$ then $\varphi \notin \text{Cl}(\alpha) \cup \text{Cl}(\beta)$. Now

$$\begin{aligned} \{\varphi \mid \varphi \bowtie \alpha\mathbf{U}\beta\} &= \{\varphi \in \text{Cl}(\alpha\mathbf{U}\beta) \mid \alpha\mathbf{U}\beta \triangleleft \varphi\} \\ &= \{\varphi \in \{\alpha\mathbf{U}\beta, \mathbf{X}(\alpha\mathbf{U}\beta), \alpha \wedge \mathbf{X}(\alpha\mathbf{U}\beta), \beta \vee (\alpha \wedge \mathbf{X}(\alpha\mathbf{U}\beta))\} \mid \alpha\mathbf{U}\beta \triangleleft \varphi\} \\ &\quad \cup \{\varphi \in \text{Cl}(\alpha) \mid \alpha\mathbf{U}\beta \triangleleft \varphi\} \cup \{\varphi \in \text{Cl}(\beta) \mid \alpha\mathbf{U}\beta \triangleleft \varphi\} \\ &= \{\alpha\mathbf{U}\beta, \mathbf{X}(\alpha\mathbf{U}\beta), \alpha \wedge \mathbf{X}(\alpha\mathbf{U}\beta), \beta \vee (\alpha \wedge \mathbf{X}(\alpha\mathbf{U}\beta))\} \cup \emptyset, \end{aligned}$$

thus the equivalence class of $\alpha\mathbf{U}\beta$ is of the desired form. The proof for $\alpha\mathbf{R}\beta$ is dual. \square

We will refer to the above types of equivalence classes as **U-cells** and **R-cells** respectively. The following proposition combines the previous two results and gives us a nice way to classify infinite traces: traces that end up in a **U-cell** are “bad” because they contradict the finiteness of the unfolding of **U**, whereas traces that end up in a **R-cell** are “good” because **R**-formulas may be unfolded infinitely often.

Proposition 3.7. *Every complete infinite trace ends up in either a **U-cell** or an **R-cell**.*

Proof. Let $\varphi_0, \varphi_1, \dots$ be a complete infinite trace. By Proposition 3.5 there is a point after which all φ_i belong to the same equivalence class of \bowtie . As the trace is non-stalling, this class contains more than one element, hence is either a **U-cell** or an **R-cell** by Proposition 3.6. \square

Note that as long as the trace is non-stalling, it does not matter if we “skip” any formulas; for example the trace that repeats $p\mathbf{U}q \triangleright \mathbf{X}(p\mathbf{U}q) \triangleright (p \wedge \mathbf{X}(p\mathbf{U}q)) \triangleright p\mathbf{U}q$ ends up in the equivalence class of $p\mathbf{U}q$. In fact this trace is cofinal with respect to the direct trace that repeats $p\mathbf{U}q \triangleright (q \vee (p \wedge \mathbf{X}(p\mathbf{U}q))) \triangleright (p \wedge \mathbf{X}(p\mathbf{U}q)) \triangleright \mathbf{X}(p\mathbf{U}q) \triangleright p\mathbf{U}q$, which clearly stays in the equivalence class of $p\mathbf{U}q$.

Definition 3.8. A trace $\varphi_0, \varphi_1, \dots$ is *cofinal* with respect to another trace ψ_0, ψ_1, \dots if for every i there is $j \geq i$ with $\psi_i \triangleright \varphi_j$. \triangleleft

Mutually cofinal traces are guaranteed to end up in the same cell, as is ensured by the following proposition. As a result, any two complete traces that are mutually cofinal either both end up in a **U-cell** or both end up in a **R-cell**.

Proposition 3.9. *Let $\varphi_0, \varphi_1, \dots$ and ψ_0, ψ_1, \dots be mutually cofinal traces, then there is k such that $\varphi_i \bowtie \psi_j$ for all $i, j \geq k$.*

Proof. Let $\varphi_0, \varphi_1, \dots$ and ψ_0, ψ_1, \dots be mutually cofinal traces. Let l and m be such that $\varphi_i \bowtie \varphi_j$ for all $i, j \geq l$ and $\psi_i \bowtie \psi_j$ for all $i, j \geq m$, as given by Proposition 3.5. Take $k := \max\{l, m\}$. Let $i, j \geq k$, then by cofinality of the first trace there is a $i' \geq j$ such that $\varphi_j \triangleright \psi_{i'}$, and by cofinality of the second trace there is $j' \geq i'$ such that $\varphi_{i'} \triangleright \psi_{j'}$. Now $j, j' \geq k \geq m$ hence $\psi_j \bowtie \psi_{j'}$, which gives $\psi_j \triangleright \varphi_{i'} \triangleright \psi_{j'} \triangleright \psi_j$ hence $\psi_j \bowtie \varphi_{i'}$. Also $i, i' \geq k \geq l$ hence $\varphi_i \bowtie \varphi_{i'}$. Thus $\varphi_i \bowtie \psi_j$. \square

Before we define the evaluation game for CTL^* we take a look at the closure of CTL. As said, for every CTL formula $\varphi\text{U}\psi$ its unfolding $\psi \vee (\varphi \wedge \text{X}(\varphi\text{U}\psi))$ must be added to the closure since it is not in CTL itself, and the same holds for the unfolding of every R-formula in CTL. Luckily this is all that needs to be added; we do not have to worry about an infinite chain of unfoldings that become less and less “CTL-like”. The only extra formulas in the closure of CTL are elements of the equivalence classes of CTL formulas.

Proposition 3.10. *For every $\varphi \in \text{Cl}(\text{CTL})$ there is $\psi \in \text{CTL}$ such that $\varphi \bowtie \psi$.*

Proof. First note that this holds for every derivative of a CTL formula. For if $\varphi \in \nabla(\psi)$ and $\psi \in \text{CTL}$, then either φ is a subformula of ψ hence $\varphi \in \text{CTL}$, or $\psi = \alpha\text{U}\beta$ and $\varphi = \beta \vee (\alpha \wedge \text{X}\psi) \bowtie \psi$, or $\psi = \alpha\text{R}\beta$ and $\varphi = \alpha \wedge (\beta \vee \text{X}\psi) \bowtie \psi$.

We can construct $\text{Cl}(\text{CTL})$ from CTL by taking $\text{Cl}(\text{CTL}) = \bigcup_{n=0}^{\infty} \Gamma_n$, where $\Gamma_0 = \text{CTL}$ and where Γ_{n+1} consist of all the derivatives of Γ_n . Suppose that for all $\varphi \in \Gamma_n$ there is $\varphi' \in \text{CTL}$ with $\varphi \bowtie \varphi'$; certainly this holds for Γ_0 . Let $\varphi \in \Gamma_{n+1}$, then φ is the derivative of some $\psi \in \Gamma_n$. The induction hypothesis then states that there is $\psi' \in \text{CTL}$ such that $\psi \bowtie \psi'$. If $\psi = \psi'$ then φ is a derivative of a CTL formula and we are done. If $\psi \neq \psi'$ then ψ and ψ' both belong to the same non-trivial equivalence class, which by Proposition 3.6 is either a U-cell or an R-cell. Looking at the contents of the equivalence class of a formula $\alpha\text{U}\beta$, we see that only $\alpha\text{U}\beta$ itself can be a CTL formula, and thus if ψ' belongs to the equivalence class of $\alpha\text{U}\beta$ then $\psi' = \alpha\text{U}\beta$. This means that $\psi' = \alpha\text{U}\beta$ or $\psi' = \alpha\text{R}\beta$. We find that either φ is α or β , hence a subformula of ψ' and thus in CTL, or $\varphi \bowtie \psi'$. Thus we have shown that for all n and all $\varphi \in \Gamma_n$, there is $\varphi' \in \text{CTL}$ with $\varphi \bowtie \varphi'$, and we conclude that it holds for all $\varphi \in \text{Cl}(\text{CTL})$. \square

Because CTL^* has both state-formulas and path-formulas, the board of the evaluation game for a CTL^* formula ξ consists of two parts, one $S \times \text{Cl}_{\Sigma}(\xi)$ and the other $\Pi(\mathbb{S}) \times \text{Cl}_{\Pi}(\xi)$. The match goes from the first part to the second part whenever a formula $\text{E}\psi$ or $\text{A}\psi$ with $\psi \in \text{CTL}_{\Pi}^*$ is reached, and it goes back whenever a formula $\varphi \in \text{CTL}_{\Sigma}^*$ is reached. Technically the formula $p \wedge q$ is both a state-formula and the conjunction of two path-formulas, but because $p \wedge q$ as a state-formula is still a conjunction, this does not lead to any fundamental problems.

Definition 3.11. Let ξ be a CTL^* formula and let $\mathbb{S} = (S, R, V)$ be a transition system. The *evaluation game for ξ on \mathbb{S}* , denoted $\mathcal{E}(\xi, \mathbb{S})$, is played on the board

$$S \times \text{Cl}_{\Sigma}(\xi) \cup \Pi(\mathbb{S}) \times \text{Cl}_{\Pi}(\xi)$$

according to the rules given in Figure 3.1. Every match induces a direct trace through CTL^* . Now Proposition 3.7 tells us that all infinite matches end up in either a U-cell or an R-cell; in the former case \forall wins, in the latter case \exists wins. \triangleleft

In the evaluation game of a μML formula ξ , the rule for a position (s, x) as given in Figure 2.1 depends on ξ by referencing $\lambda_{\xi}(x)$. If we look closely at the rules as given in Figure 3.1, however, it becomes clear that these rules are completely independent of ξ . In

position	player	admissible moves	
(s, ℓ)	\forall	\emptyset	if $s \Vdash_V \ell$
(s, ℓ)	\exists	\emptyset	if $s \not\Vdash_V \ell$
$(s, \varphi_1 \vee \varphi_2)$	\exists	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \varphi_1 \wedge \varphi_2)$	\forall	$\{(s, \varphi_1), (s, \varphi_2)\}$	
$(s, \mathbf{E}\psi)$	\exists	$\{(\pi, \psi) \mid \pi \in \Pi(\mathbb{S}, s)\}$	
$(s, \mathbf{A}\psi)$	\forall	$\{(\pi, \psi) \mid \pi \in \Pi(\mathbb{S}, s)\}$	
(π, φ)	-	$\{(\pi(0), \varphi)\}$	if $\varphi \in \text{CTL}_\Sigma^*$
$(\pi, \psi_1 \vee \psi_2)$	\exists	$\{(\pi, \psi_1), (\pi, \psi_2)\}$	
$(\pi, \psi_1 \wedge \psi_2)$	\forall	$\{(\pi, \psi_1), (\pi, \psi_2)\}$	
$(\pi, \mathbf{X}\psi)$	-	$\{(\pi^1, \psi)\}$	
$(\pi, \psi_1 \mathbf{U}\psi_2)$	-	$\{(\pi, \psi_2 \vee (\psi_1 \wedge \mathbf{X}(\psi_1 \mathbf{U}\psi_2)))\}$	
$(\pi, \psi_1 \mathbf{R}\psi_2)$	-	$\{(\pi, \psi_2 \wedge (\psi_1 \vee \mathbf{X}(\psi_1 \mathbf{R}\psi_2)))\}$	

Figure 3.1: The rules for the evaluation game of CTL* formulas.

fact we can see an evaluation match of a CTL* formula ξ on a transition system \mathbb{S} as a match played on a much larger board, $S \times \text{CTL}_\Sigma^* \cup \Pi(\mathbb{S}) \times \text{CTL}_\Pi^*$, that only depends on \mathbb{S} . This gives evaluation games for CTL* a useful context-independence.

Proposition 3.12. *Let \mathbb{S} be a transition system and let φ, ψ be CTL* formulas. If a position p is both on the board of $\mathcal{E}(\varphi, \mathbb{S})$ and on the board of $\mathcal{E}(\psi, \mathbb{S})$, then p is a winning position for \exists in the game $\mathcal{E}(\varphi, \mathbb{S})$ iff p is a winning position for \exists in the game $\mathcal{E}(\psi, \mathbb{S})$.*

Proof. Let \mathbb{S} , φ and ψ be as such. The positions appearing on both boards are exactly the positions in the subboard

$$S \times (\text{Cl}_\Sigma(\varphi) \cap \text{Cl}_\Sigma(\psi)) \cup \Pi(\mathbb{S}) \times (\text{Cl}_\Pi(\varphi) \cap \text{Cl}_\Pi(\psi)).$$

For any position (s, χ) or (π, χ) in this subboard we have $\text{Cl}(\chi) \subseteq \text{Cl}(\varphi) \cap \text{Cl}(\psi)$, hence any admissible move from such a position leads to another position in the subboard. Therefore a match that enters the subboard never leaves it. The result follows since the rules given in Figure 3.1 and the winning condition for infinite matches, as given by Proposition 3.6, don't depend on the initial formula, be it φ or ψ . \square

Before we can use evaluation game semantics for CTL*, we first need to reconcile these semantics with the relational semantics defined in section 2.1. We will make implicit use of this theorem throughout this paper.

Theorem 3.13 (Adequacy of Game Semantics for CTL*). *Let \mathbb{S} be a transition system. Let $\varphi \in \text{CTL}_\Sigma^*$ and $s \in S$, then $\mathbb{S}, s \Vdash \varphi$ iff $(s, \varphi) \in \text{Win}_\exists(\mathcal{E}(\varphi, \mathbb{S}))$. Let $\psi \in \text{CTL}_\Pi^*$ and $\pi \in \Pi(\mathbb{S})$, then $\mathbb{S}, \pi \Vdash \psi$ iff $(\pi, \psi) \in \text{Win}_\exists(\mathcal{E}(\psi, \mathbb{S}))$.*

Proof. By induction on the structure of the formula. We highlight the case where $\psi = \alpha \mathbf{U} \beta$. Let $\pi \in \Pi(\mathbb{S})$.

Suppose $\mathbb{S}, \pi \Vdash \alpha \mathbf{U} \beta$, then there is k with $\mathbb{S}, \pi^k \Vdash \beta$ and $\mathbb{S}, \pi^i \Vdash \alpha$ for all $i < k$. Using the induction hypothesis \exists has winning strategies for $\mathcal{E}(\beta, \mathbb{S}) @ (\pi^k, \beta)$ and for $\mathcal{E}(\alpha, \mathbb{S}) @ (\pi^i, \alpha)$ for each i . This means that (π^k, β) and each of the (π^i, α) are winning positions in their respective games, and by Proposition 3.12 they are also winning positions in the game $\mathcal{E}(\alpha \mathbf{U} \beta, \mathbb{S})$. Now $(\pi^k, \beta \vee (\alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta)))$ is a winning position for \exists , as she is free to choose the left disjunct. Therefore $(\pi^k, \alpha \mathbf{U} \beta)$ is a winning position for \exists .

If $(\pi^{i+1}, \alpha \mathbf{U} \beta)$ and (π^i, α) are winning positions for \exists , then so is $(\pi^i, \mathbf{X}(\alpha \mathbf{U} \beta))$ and hence so is $(\pi^i, \alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta))$, since any choice of \forall leads to a winning position. This in turn gives that $(\pi^i, \beta \vee (\alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta)))$ is winning for \exists , as she is free to choose the right disjunct. Therefore so is $(\pi^i, \alpha \mathbf{U} \beta)$. By backwards induction we get that $(\pi^0, \alpha \mathbf{U} \beta)$ is a winning position for \exists . Thus \exists has a winning strategy for $\mathcal{E}(\alpha \mathbf{U} \beta, \mathbb{S}) @ (\pi, \alpha \mathbf{U} \beta)$.

Suppose \exists has a winning strategy for $\mathcal{E}(\alpha \mathbf{U} \beta, \mathbb{S}) @ (\pi, \alpha \mathbf{U} \beta)$. This means that $(\pi, \alpha \mathbf{U} \beta)$ is a winning position for \exists , hence so is $(\pi, \beta \vee (\alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta)))$. If $(\pi^i, \beta \vee (\alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta)))$ is a winning position for \exists , then so is either (π^i, β) or $(\pi^i, \alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta))$; in the latter case so are (π^i, α) , $(\pi^i, \mathbf{X}(\alpha \mathbf{U} \beta))$, $(\pi^{i+1}, \alpha \mathbf{U} \beta)$ and $(\pi^{i+1}, \beta \vee (\alpha \wedge \mathbf{X}(\alpha \mathbf{U} \beta)))$. Because a match that would go through infinitely many $(\pi^i, \alpha \mathbf{U} \beta)$ would be lost by \exists , there must be a least k for which (π^k, β) is a winning position. By the induction hypothesis and Proposition 3.12, we have $\mathbb{S}, \pi^k \Vdash \beta$ and $\mathbb{S}, \pi^i \Vdash \alpha$ for all $i < k$. Thus $\mathbb{S}, \pi \Vdash \alpha \mathbf{U} \beta$. \square

As with the evaluation games for μML and the acceptance games for modal automata, it will prove extremely useful to fully embrace the dual nature of these game; therefore we need to have $\mathbb{S}, s \not\Vdash \varphi$ whenever $(s, \varphi) \in \text{Win}_\forall(\mathcal{E}(\varphi, \mathbb{S}))$. This is given by the determinacy.

Proposition 3.14. *Evaluation games for CTL* formulas enjoy positional determinacy.*

Proof. The winning condition from Definition 3.11 is equivalent to a parity winning condition, where \mathbf{U} -formulas have priority 1 and other formulas priority 0. The result therefore follows from Theorem 2.18. \square

3.2 Dominated formulas

As mentioned at the start of this chapter, a crucial challenge in the translation from CTL* to modal automata and the modal μ -calculus is working out how to translate path-formulas. This is because the semantics of CTL* use paths as well as states, whereas the semantics of automata and the semantics of μML only use states. As an example, consider the CTL* formula $\text{EGF}p$, which states that there exists a path on which p is true infinitely often. In the evaluation game for this formula, \exists is first tasked to find a path π and the game then continues to evaluate $\text{GF}p$ on this path π . In the evaluation game of a modal formula, it is not possible for \exists to “commit” to a path in this way; whenever a subformula $\diamond\psi$ is reached, \exists is free to choose any reachable state. Simply dropping the commitment comes down to evaluating the CTL formula $\text{EGEF}p$ instead. Since CTL is significantly weaker than CTL*, this will not do.

Instead, we will give \exists more power in the evaluation games and acceptance games that correspond to formulas $\text{E}\psi$. She will be able to dictate the flow of the match, in

the sense that she decides which states of the transition system the match goes through. Note that in games with modal formulas, choices by \forall are made at formulas $\Box\varphi$ and at formulas $\varphi \wedge \psi$. As an extreme, outright banning \Box and \wedge would give \exists total control. If we look at a formula such as $\text{EGF}(p \wedge q)$, however, we see that this can't work; surely any equivalent μML formula needs to contain a conjunction. Still, we will use formulas that are “dominated” by \exists and where the use of \Box and \wedge are limited in certain ways. Likewise, modal formulas used in games that correspond to formulas $\mathbf{A}\psi$ will be dominated by \forall and will make limited use of \Diamond and \vee . For now we will be interested in modal automata whose transition terms are dominated.

Definition 3.15. A $\text{ML}_1(A)$ formula is X -free for some $X \subseteq A$ if it is in $\text{ML}_1(A \setminus X)$. It is \exists -dominated with respect to X if every subformula $\Box\varphi$ is X -free and for every subformula $\varphi \wedge \psi$ either φ or ψ is X -free. It is \forall -dominated with respect to X if every subformula $\Diamond\varphi$ is X -free and for every subformula $\varphi \vee \psi$ either φ or ψ is X -free. It is *dominated* with respect to X if it is either \exists -dominated or \forall -dominated. \triangleleft

It may be clear that if a formula is X -free, then it is also Y -free for every $Y \subseteq X$. As such, any formula that is \exists -dominated with respect to X is also \exists -dominated with respect to every $Y \subseteq X$. Also note that some formulas, in particular X -free formulas, can be simultaneously \exists -dominated and \forall -dominated with respect to X ; this is only a linguistic issue, and we will simply pick one of the two players if necessary. Now let us define what it means for an automaton to have *dominated clusters*.

Definition 3.16. A (generalized) cluster C of a modal automaton \mathbb{A} is \exists -dominated if for every $c \in C$, $\Delta(c)$ is \exists -dominated with respect to C . It is \forall -dominated if for every $c \in C$, $\Delta(c)$ is \forall -dominated with respect to C . It is *dominated* if it is either \exists -dominated or \forall -dominated. \triangleleft

Importantly, for a cluster C to be dominated it is not enough that the transition term of every state in C is dominated with respect to C ; the transition terms must all be dominated by the same player, be it \exists or \forall . Note that if a generalized cluster C is dominated by a player, then any subcluster of C is also dominated by that player. Now for guarded modal automata with dominated clusters, the transition terms can be written in a special form.

Proposition 3.17. For every $\text{gML}_1(A)$ formula φ that is dominated with respect to a set $C = \{c_1, \dots, c_n\} \subseteq A$ there are formulas $\varphi_0, \varphi_1, \dots, \varphi_n \in \text{gML}_1(A \setminus C)$ such that

$$\varphi \equiv \varphi_0 \vee (\varphi_1 \wedge \Diamond c_1) \vee \dots \vee (\varphi_n \wedge \Diamond c_n)$$

if it is \exists -dominated, or

$$\varphi \equiv \varphi_0 \wedge (\varphi_1 \vee \Box c_1) \wedge \dots \wedge (\varphi_n \vee \Box c_n)$$

if it is \forall -dominated.

Proof. Let φ be guarded and \exists -dominated with respect to C ; the case where φ is \forall -dominated is dual. We can write φ in a disjunctive normal form and get $\varphi \equiv \alpha_1 \vee \dots \vee \alpha_m$ where $\alpha_i = \beta_{i,1} \wedge \dots \wedge \beta_{i,k_i}$. Because φ is \exists -dominated we know that for every i at most one $\beta_{i,j}$ contains an occurrence of an element of C , and because it is guarded such occurrences are \diamond -guarded. Thus for every i we can either find a unique $j_i \in \{1, \dots, n\}$ such that α_i contains a conjunct $\diamond c_{j_i}$, or set $j_i = 0$. For every $i \leq m$ create α'_i from $\alpha_i \wedge \top$ by removing the conjunct $\diamond c_{j_i}$, then α'_i contains no occurrences of $c \in C$, hence $\alpha'_i \in \text{gML}_1(A \setminus C)$. For $j \in \{1, \dots, n\}$ let φ_j be the disjunction of all the α'_i for which $j_i = j$, or \perp if there are no such i . Let φ_0 be the disjunction of all α_i where $j_i = 0$, or \perp . Now we find $\Delta(a) \equiv \varphi_0 \vee (\varphi_1 \wedge \diamond c_1) \vee \dots \vee (\varphi_n \wedge \diamond c_n)$. \square

This special form further illustrates how one player controls the flow of the game. In the case of an \exists -dominated cluster C , it says that \exists can pick a candidate state $c \in C$, and then \forall can choose either to go to c and have \exists pick a successor, or to leave the cluster. In this way, \exists decides the path through the transition system that the match will follow as long as the match remains in the cluster.

3.3 Cluster-path acceptance

We started this chapter by noting that acceptance games for modal automata do not operate on paths. We then saw that for modal automata with dominated clusters, one player can dictate the path that the acceptance match operators on as long as the match remains inside a cluster. In this section we solidify this idea with an alternate game semantics for modal automata. In the “cluster-path acceptance game” for a cluster C and a path π , the match is forced to stay on the path π for as long as the match remains in the cluster C . As a result, the modality \diamond and \square are no longer decision points for \exists and \forall , instead behaving somewhat as the X operator from CTL*. When the cluster is left, however, the path is discarded and the match continues as normal.

Definition 3.18. Let \mathbb{A} be a modal automaton and let C be a (generalized) cluster of \mathbb{A} . Let \mathbb{S} be a transition system and let π be a path through \mathbb{S} . The *cluster-path acceptance game for \mathbb{A} on \mathbb{S} with respect to C and π* , denoted $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$, is played on the board

$$(\mathbb{N} \times \text{ML}_1(A)) \cup (S \times A)$$

according to the rules given in Figure 3.2. Every infinite match induces an infinite sequence of states from A by looking at the positions of the form (i, a) where $a \in C$. An infinite match is won by \exists iff the acceptance condition of \mathbb{A} accepts this sequence. \triangleleft

Clearly this game poses a restriction for the players. Since in \exists -dominated clusters \forall did not have much control over the match to begin with, \exists is restricted the most. Thus it makes sense that if she is able to win with these restrictions in place, then she can also win without them. The following proposition states this.

position	player	admissible moves	
(i, a)	-	$\{(i, \Delta(a))\}$	if $a \in C$
(i, a)	-	$\{(\pi(i), a)\}$	if $a \notin C$
(i, ℓ)	\forall	\emptyset	if $\pi(i) \Vdash_V \ell$
(i, ℓ)	\exists	\emptyset	if $\pi(i) \not\Vdash_V \ell$
$(i, \varphi_1 \vee \varphi_2)$	\exists	$\{(i, \varphi_1), (i, \varphi_2)\}$	
$(i, \varphi_1 \wedge \varphi_2)$	\forall	$\{(i, \varphi_1), (i, \varphi_2)\}$	
$(i, \diamond a)$	-	$\{(i+1, a)\}$	if $a \in C$
$(i, \diamond a)$	\exists	$\{(t, a) \mid \pi(i)Rt\}$	if $a \notin C$
$(i, \square a)$	-	$\{(i+1, a)\}$	if $a \in C$
$(i, \square a)$	\forall	$\{(t, a) \mid \pi(i)Rt\}$	if $a \notin C$
(s, a)	\forall	\emptyset	if $(s, a) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$
(s, a)	\exists	\emptyset	if $(s, a) \notin \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$

Figure 3.2: The rules for the cluster-path acceptance game of modal automata.

Proposition 3.19. *Let \mathbb{A} be a modal automaton and let C be a dominated (generalized) cluster of \mathbb{A} . Let \mathbb{S} be a transition system. For all $a \in C$ and $s \in S$:*

1. *if C is \exists -dominated and \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ for some path π starting at s , then \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$;*
2. *if C is \forall -dominated and \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ for some path π starting at s , then \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$.*

Proof. We will prove (1.); the proof for (2.) is dual.

Let C be \exists -dominated, let $a \in C$ and let $s \in S$. Suppose that π is a path starting at s and that f is a winning strategy of \exists for the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. While \exists and \forall are playing a match m of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$, \exists will play a shadow match m' of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ such that every shadow position of the form (k_i, φ_i) corresponds to the real position $(\pi(k_i), \varphi_i)$, and a shadow position of the form (t, b) with $b \notin C$ corresponds to the real position (t, b) .

Since m starts at (s, a) , m' starts at $(0, a)$ and $\pi(0) = s$, we can start both matches in this way. Whenever the matches are at positions $(\pi(k), c)$ and (k, c) with $c \in C$, they will necessarily continue to $(\pi(k), \Delta(c))$ and $(k, \Delta(c))$. Whenever they are at $(\pi(k), \ell)$ and (k, ℓ) , both matches end; it may be clear that the winner of both matches is the same. Whenever they are at $(\pi(k), \varphi_1 \vee \varphi_2)$ and $(k, \varphi_1 \vee \varphi_2)$, the strategy f of \exists prescribes a move (k, φ_i) of the shadow match. Here \exists performs the move $(\pi(k), \varphi_i)$ in the real match. Whenever they are at $(\pi(k), \varphi_1 \wedge \varphi_2)$ and $(k, \varphi_1 \wedge \varphi_2)$, any move $(\pi(k), \varphi_i)$ that \forall makes in the real match can be mimicked in the shadow match with (k, φ_i) . Whenever they are at $(\pi(k), \diamond c)$ and $(k, \diamond c)$ with $c \in C$, the shadow match necessarily continues to $(k+1, c)$. Here \exists performs the move $(\pi(k+1), c)$ in the real match. Because C is \exists -dominated, positions of the form $\square c$ with $c \in C$ never occur.

Whenever they are at $(\pi(k), b)$ and (k, b) where $b \notin C$, the shadow match necessarily continues at $(\pi(k), b)$ and then ends; because f is winning, it ends in a victory for \exists hence

$(\pi(k), b) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$. Thus \exists has some strategy g with which she can continue the real match m to a victorious conclusion. Whenever they are at $(\pi(k), \diamond b)$ and $(k, \diamond b)$ where $b \notin C$, the shadow match continues to some (t, b) with $\pi(k)Rt$ as prescribed by f , and then ends; again this means $(t, b) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$. Here \exists performs the move (t, b) in the real match as well and will eventually win. Whenever they are at $(\pi(k), \square b)$ and $(k, \square b)$ where $b \notin C$, any move (t, b) that \forall makes in the real match can be mimicked in the shadow match with (t, b) . The shadow match then ends with \exists winning, hence $(t, b) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{S}))$. Thus any move \forall can make leads the real match to a position that is winning for \exists .

In this way m' can be simulated by \exists while m is being played. Because m' is consistent with f , \exists wins m' . The shadow match can end in one of three ways. If m' is finite and ends in a true literal, then m also ends in that literal and \exists wins m . If m' is finite and ends in a position (t, b) with $b \in C$, this position must be winning for \exists ; as m also goes through (t, b) , \exists wins m . If m' is infinite then its i -th position is of the form (k_i, φ_i) and the i -th position of m is of the form $(\pi(k_i), \varphi_i)$. Therefore the induced sequences are identical. Since \exists wins m' , this sequence is accepted by the acceptance condition of \mathbb{A} , and thus \exists wins m .

This shows that \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$. \square

The true strength of dominated clusters lies in the converse of the Proposition 3.19. The following proposition implies that whenever \exists enters an \exists -dominated cluster, she actually can commit to a single path π . This will aid us in creating automata for formulas $\text{E}\psi$; we construct a cluster for ψ , and then show that she can win an acceptance game starting in this cluster if there is a path for which she can win the cluster-path acceptance game.

Proposition 3.20. *Let \mathbb{A} be a modal automaton and let C be a dominated (generalized) cluster of \mathbb{A} . Let \mathbb{S} be a transition system. For all $a \in C$ and $s \in S$:*

1. *if C is \exists -dominated then \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$ iff she has one for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ for some path π starting at s ;*
2. *if C is \forall -dominated then \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$ iff he has one for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ for some path π starting at s ;*

Proof. We will prove (1.); the proof for (2.) is dual. Note that one direction is already given by Proposition 3.19.

Let C be a \exists -dominated cluster, let $a \in C$ and let $s \in S$. Suppose that f is a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$. We need to find a path π such that \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. To do this we play a mock match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$ that we keep inside the cluster C for as long as possible. Whenever \exists has to make a choice, she chooses according to her winning strategy f . Whenever the mock match arrives at a position $(t, \varphi_1 \wedge \varphi_2)$ at least one of the conjuncts does not contain any occurrences from states in C , because C is \exists -dominated. Here mock- \forall chooses whichever conjunct has an occurrence from a state in C ; the match ends prematurely if neither conjunct does. Whenever the mock match arrives at a position $(t, \square b)$ we have $b \notin C$ because C is \exists -dominated; the

match ends prematurely. In this way a possibly partial mock match m' is played. We construct π from this m' by first taking $\pi(0) = s$, and whenever a position $(\pi(i), \diamond b)$ occurs it is followed by some (t, b) , where we define $\pi(i + 1) := t$. If the mock match m' was partial and the last position was $\pi(k)$, then we can extend π to a proper path by appending an arbitrary path from $\Pi(\mathbb{S}, \pi(k))$.

We note that if m'' is an initial segment of m' then this m'' is consistent with f hence its last position is winning for \exists . As a result, every position in m' is a winning position for \exists . Furthermore if m'' is an initial segment of m' that ends in a position where \forall has to make a move, then any move by \forall brings the match to a winning position for \exists .

We are left to prove that \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. Her strategy is such that whenever m is a partial segment of an infinite match of this game, replacing each (k, φ) with $(\pi(k), \varphi)$ yields an initial segment of m' . Certainly this holds for $(0, a)$ which yields (s, a) , an initial segment of m' . Whenever the real match is at a position (k, c) with $c \in C$ and the mock match at the position $(\pi(k), c)$, the real match and the mock match necessarily continue to $(k, \Delta(c))$ and $(\pi(k), \Delta(c))$ respectively. Whenever the matches are at $(k, \diamond c)$ and $(\pi(k), \diamond c)$ with $c \in C$, the real match necessarily continues to $(k + 1, c)$ and the mock match continues to $(\pi(k + 1), c)$ by construction of π . Because C is \exists -dominated, positions of the form $(k, \square c)$ with $c \in C$ cannot occur.

Whenever the matches are at (k, ℓ) and $(\pi(k), \ell)$, \exists wins the real match because she won the mock match. Whenever the matches are at (k, b) and $(\pi(k), b)$ with $b \notin C$, the real match necessarily continues to $(\pi(k), b)$ which is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})$. Therefore \exists wins both matches. Whenever the matches are at $(k, \diamond b)$ and $(\pi(k), \diamond b)$ with $b \notin C$, the strategy f prescribes a move (t, b) which is also admissible in the real match. As (t, b) is a winning position for \exists in $\mathcal{A}(\mathbb{A}, \mathbb{S})$, \exists wins both matches. Whenever the matches are at $(k, \square b)$ and $(\pi(k), \square b)$ with $b \notin C$, any move (t, b) that \forall can make in the real match is admissible in the partial mock match, hence (t, b) is a winning position for \exists in $\mathcal{A}(\mathbb{A}, \mathbb{S})$ and \exists wins both matches.

Whenever the matches are at $(k, \varphi_1 \vee \varphi_2)$ and $(\pi(k), \varphi_1 \vee \varphi_2)$ respectively, the strategy f prescribes a move $(\pi(k), \varphi_i)$. Here \exists chooses (k, φ_i) . Whenever the matches are at $(k, \varphi_1 \wedge \varphi_2)$ and $(\pi(k), \varphi_1 \wedge \varphi_2)$ respectively and \forall performs the move (k, φ_i) in the real match, either the mock match continues to $(\pi(k), \varphi_i)$ or φ_i does not contain occurrences from states in C . In the latter case we have that the real match will stay at k and end after a finite number of moves because no position $(k, \diamond c)$ or $k, \square c$ with $c \in C$ can be reached.

In this manner \exists can play the match m of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. We have seen that if m is finite, then \exists wins. If m is infinite, then each position (k, φ) corresponds to $(\pi(k), \varphi)$ in m' , hence the induced sequences are identical for both matches. Because \exists won m' this sequence is accepted by the acceptance condition of \mathbb{A} , hence \exists wins m . \square

When looking at Proposition 3.17, we noted that for an \exists -dominated cluster \exists picks not only the successor in the transition system, but also the next state of the automaton. The following proposition extrapolates this further; \exists can in fact plan out the entire match of the cluster-path acceptance game in advance, and this plan will prove correct for as long as the actual match remains in the cluster.

Proposition 3.21. *Let \mathbb{A} be a guarded modal automaton and let C be a dominated cluster of \mathbb{A} . Let \mathbb{S} be a transition system. For all $a \in C$ and $\pi \in \Pi(\mathbb{S})$:*

1. *if C is \exists -dominated then for every winning strategy f of \exists for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ there is a function $\epsilon : \mathbb{N} \rightarrow C$ such that for all positions (i, b) that occur in a match consistent with f , either $b = \epsilon(i)$ or $b \notin C$.*
2. *if C is \forall -dominated then for every winning strategy of \forall there is such a function.*

Proof. Let \mathbb{A} , C , \mathbb{S} , a and π be as such. Let C be \exists -dominated; the case where C is \forall -dominated is dual. Let f be a winning strategy of \exists for the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. To construct ϵ , \exists plays a possibly partial mock match that is consistent with f . If the mock match reaches a position $(i, \varphi_1 \vee \varphi_2)$, then f instructs \exists to move either to (i, φ_1) or to (i, φ_2) . If the mock match reaches a position $(i, \varphi_1 \wedge \varphi_2)$, then because C is \exists -dominated at most one conjunct contains occurrences of states in C . Now if φ_1 is this conjunct we have mock- \forall move to (i, φ_1) ; otherwise we have him move to (i, φ_2) . In this way the mock match is played out to completion or until a position $(i, \diamond b)$ or $(i, \square b)$ with $b \notin C$ is reached. If a position (i, c) is followed by a position (j, d) with $i, j \in \mathbb{N}$ and $c, d \in C$, then $i < j$ because \mathbb{A} is guarded. For every $i \in \mathbb{N}$ for which a position (i, c) exists in the mock match, define $\epsilon(i) := c$. For all other i define $\epsilon(i) := a$.

Now \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$ that is consistent with f . The first positions of the real match and the mock match are the same: $(0, a)$. Suppose the k -th positions of the real match and the mock match are the same. If \exists has to move, then in both matches she will move according to her strategy f hence also the $(k+1)$ -th positions of both matches are the same. If the move is automatic, then the result is the same in both matches. If \forall has to move and the k -th position is $(i, \varphi_1 \wedge \varphi_2)$ then either \forall moves to the same position as mock- \forall did, or \forall moves to a position (i, φ) where φ contains no occurrences of states in C . Since C is a cluster, this means no states from C will occur for the rest of the real match. If the mock match ends prematurely because a position $(i, \diamond b)$ or $(i, \square b)$ with $b \notin C$ is reached, then neither match will contain any more states from C . We conclude that if the k -th position of the real match is (i, c) with $c \in C$, then the two matches are identical up to the first k moves; in particular $c = \epsilon(i)$. \square

Chapter 4

An automaton construction for CTL*

In the following section, we will construct unguarded modal Muller automata for CTL* formulas. The automata will be given a parity condition and made guarded in section 4.2.

It will prove convenient to work with *Dam terms*, which are quantified sets of formulas $Q\Phi$. We will use sequential notation for $Q\Phi$, writing $Q(\Phi, \varphi)$ instead of $Q(\Phi \cup \{\varphi\})$. Sets are interpreted conjunctively if quantified by E and disjunctively if quantified by A. Thus given a transition system \mathbb{S} and a state $s \in S$ we will write $\mathbb{S}, s \Vdash E\Phi$ instead of $\mathbb{S}, s \Vdash E(\varphi_1 \wedge \dots \wedge \varphi_n)$ if $\Phi = \{\varphi_1, \dots, \varphi_n\}$. Every CTL* _{Σ} formula ξ has an equivalent Dam term; if $\xi = Q\psi$ then it is $Q(\psi)$, and otherwise we will take $E(\xi)$.

4.1 Automaton construction

The transition terms of the automaton will be guided by *Dam rules*. Dam rules exist in one of two forms: terminal and non-terminal.

$$I \frac{Q\Phi}{\ell} \quad \tau \frac{Q\Phi}{Q_1\Psi_1 \quad \dots \quad Q_n\Psi_n}$$

Here the operator τ is one of $I, \vee, \wedge, \diamond$ and \square , where I stands for the identity operation. The rules for $A\Phi$ are always dual to those of $E\Phi$. The first set of rules are *extraction* rules, in which one formula is separated from the rest.

$$\begin{array}{ccc} I \frac{E(\ell)}{\ell} & \wedge \frac{E(\Phi, \ell)}{E\Phi \quad E(\ell)} & I \frac{A(\ell)}{\ell} & \vee \frac{A(\Phi, \ell)}{A\Phi \quad A(\ell)} \\ I \frac{E(E\varphi)}{E(\varphi)} & \wedge \frac{E(\Phi, E\varphi)}{E\Phi \quad E(\varphi)} & I \frac{A(A\varphi)}{A(\varphi)} & \vee \frac{A(\Phi, A\varphi)}{A\Phi \quad A(\varphi)} \\ I \frac{E(A\varphi)}{A(\varphi)} & \wedge \frac{E(\Phi, A\varphi)}{E\Phi \quad A(\varphi)} & I \frac{A(E\varphi)}{E(\varphi)} & \vee \frac{A(\Phi, E\varphi)}{A\Phi \quad E(\varphi)} \end{array}$$

Then there are *unfolding* rules, where one formula is replaced by some other formulas. Note that in the following rules, Φ can be empty.

$$\begin{array}{c}
I \frac{E(\Phi, \varphi \wedge \psi)}{E(\Phi, \varphi, \psi)} \qquad I \frac{A(\Phi, \varphi \vee \psi)}{A(\Phi, \varphi, \psi)} \\
\vee \frac{E(\Phi, \varphi \vee \psi)}{E(\Phi, \varphi) \quad E(\Phi, \psi)} \qquad \wedge \frac{A(\Phi, \varphi \wedge \psi)}{A(\Phi, \varphi) \quad A(\Phi, \psi)} \\
\vee \frac{E(\Phi, \varphi \mathbf{U} \psi)}{E(\Phi, \psi) \quad E(\Phi, \varphi, \mathbf{X}(\varphi \mathbf{U} \psi))} \qquad \wedge \frac{A(\Phi, \varphi \mathbf{R} \psi)}{A(\Phi, \psi) \quad A(\Phi, \varphi, \mathbf{X}(\varphi \mathbf{R} \psi))} \\
\vee \frac{E(\Phi, \varphi \mathbf{R} \psi)}{E(\Phi, \psi, \varphi) \quad E(\Phi, \psi, \mathbf{X}(\varphi \mathbf{R} \psi))} \qquad \wedge \frac{A(\Phi, \varphi \mathbf{U} \psi)}{A(\Phi, \psi, \varphi) \quad A(\Phi, \psi, \mathbf{X}(\varphi \mathbf{U} \psi))}
\end{array}$$

Lastly there are two *modal* rules, where an \mathbf{X} operator is removed from every formula.

$$\begin{array}{c}
\Diamond \frac{E(\mathbf{X}\varphi_1, \dots, \mathbf{X}\varphi_n)}{E(\varphi_1, \dots, \varphi_n)} \qquad \Box \frac{A(\mathbf{X}\varphi_1, \dots, \mathbf{X}\varphi_n)}{A(\varphi_1, \dots, \varphi_n)}
\end{array}$$

Note that if a modal rule is applicable to $Q\Phi$, then no other rule is. The rules are all semantically sound, as the following two propositions show.

Proposition 4.1. *Let $I : Q\Phi \Rightarrow \ell$ be a terminal Dam rule, then $Q\Phi \equiv \ell$.*

Proof. If $I : Q\Phi \Rightarrow \ell$ is a terminal rule, then $\Phi = \{\ell\}$. Let $s \in S$. Note that by seriality, $\Pi(S, s)$ is non-empty. For a path $\pi \in \Pi(S, s)$ we have $\pi(0) = s$ and $S, \pi \Vdash \ell$ iff $S, \pi(0) \Vdash \ell$. Therefore $S, s \Vdash A\ell$ iff $S, s \Vdash \ell$ iff $S, s \Vdash E\ell$. We conclude $Q(\ell) \equiv \ell$. \square

Proposition 4.2. *Let $\tau : Q\Phi \Rightarrow Q_1\Psi_1, \dots, Q_n\Psi_n$ be a non-terminal Dam rule, then $Q\Phi \equiv \tau(Q_1\Psi_1, \dots, Q_n\Psi_n)$ where $I\xi := \xi$, $\Diamond\xi := E\mathbf{X}\xi$ and $\Box\xi := A\mathbf{X}\xi$.*

Proof. By inspection of the rules. Note in particular that $\varphi\mathbf{U}\psi \equiv \psi \vee (\varphi \wedge \mathbf{X}(\varphi\mathbf{U}\psi))$, whence the rule for $E(\Phi, \varphi\mathbf{U}\psi)$, and that $\varphi\mathbf{R}\psi \equiv \psi \wedge (\varphi \vee \mathbf{X}(\varphi\mathbf{R}\psi)) \equiv (\psi \wedge \varphi) \vee (\psi \wedge \mathbf{X}(\varphi\mathbf{R}\psi))$, whence the rule for $E(\Phi, \varphi\mathbf{R}\psi)$. \square

Furthermore, there is a strong connection between the Dam rules and the relation \triangleright defined in section 3.1. For instance, the E -rule for $\varphi\mathbf{U}\psi$ can be seen as first replacing $\varphi\mathbf{U}\psi$ with its derivative $\psi \vee (\varphi \wedge \mathbf{X}(\varphi\mathbf{U}\psi))$ and then applying the E -rules for \wedge and \vee .

Proposition 4.3. *Let $\tau : Q\Phi \Rightarrow Q_1\Psi_1, \dots, Q_n\Psi_n$ be a non-terminal Dam rule and let $i \leq n$, then $\text{Cl}(\Psi_i) \subseteq \text{Cl}(\Phi)$ and either $Q = Q_i$ or $\text{Cl}(\Psi_i) \neq \text{Cl}(\Phi)$.*

Proof. By inspection of the rules. In particular note that $Q \neq Q_i$ only happens with the extraction of a formula $Q_i\varphi \in \Phi$, where $\Psi_i = \{\varphi\}$ and where $Q_i\varphi \in \text{Cl}(\Phi) \setminus \text{Cl}(\Psi_i)$. \square

In Dam (1990), trees are built based on these rules, and Dam then analyzes the traces that run through branches of this tree. Here traces will run through sequences of continuation relations, which can be seen as bundles of traces.

Definition 4.4. A *continuation* for Φ is a non-empty binary relation Z on $\text{Cl}(\Phi)$ such that $\varphi \triangleright \psi$ for all $(\varphi, \psi) \in Z$. Let $\mathcal{Z}(\Phi)$ denote the set of all continuations on Φ . \triangleleft

Definition 4.5. A trace $\varphi_0, \varphi_1, \dots$ is said to *run through* a (finite or infinite) sequence Z_1, Z_2, \dots of continuations if $\varphi_i Z_{i+1} \varphi_{i+1}$ for all i . \triangleleft

The states of our automaton will be *quantified continuations*; they are of the form QZ where Q is either **E** or **A** and where Z is a continuation. The transition term of a state QZ is then based on a Dam rule for $Q\Phi$, where $\text{Ran}(Z) = \Phi$. If the range of Z consists of a single literal, then this rule must be terminal; in this case we will call the state QZ terminal as well. Otherwise a non-terminal rule $\tau : Q\Phi \Rightarrow Q_1\Psi_1, \dots, Q_n\Psi_n$ can be applied to QZ , and this results in the rule $\tau : QZ \Rightarrow Q_1Z_1, \dots, Q_nZ_n$, where each Z_i will have $\text{Dom}(Z_i) \subseteq \text{Ran}(Z)$ and $\text{Ran}(Z_i) = \Psi_i$.

If the range of Z contains only **X**-formulas then this rule must be modal. In that case the rule for QZ becomes $\tau : QZ \Rightarrow QZ_1$ where $Z_1 := \{(\text{X}\varphi, \varphi) \mid \text{X}\varphi \in \text{Ran}(Z)\}$ and $\tau \in \{\diamond, \square\}$ as appropriate. Otherwise there might be multiple extraction and unfolding rules that can be applied, one for each formula in $\text{Ran}(Z)$ that isn't an **X**-formula. We will assume a linear order on the formulas of CTL^* , and we will apply the rule for the greatest formula under that order.

If the rule of $Q\Phi$ is an extraction rule, then there is some $\chi \in \Phi$ which is extracted. The rule was of the form $\tau : Q\Phi \Rightarrow Q\Psi_1, Q_2\Psi_2$ where $\Psi_1 = \Phi \setminus \{\chi\}$ and the rule for QZ becomes $\tau : QZ \Rightarrow Q\{(\varphi, \varphi) \mid \varphi \in \Psi_1\}, Q_2\{(\chi, \psi) \mid \psi \in \Psi_2\}$. If the rule of $Q\Phi$ is an unfolding rule, then there is some $\chi \in \Phi$ which is unfolded. The rule was of the form $\tau : Q\Phi \Rightarrow Q\Psi_1, \dots, Q\Psi_n$ where each of the Ψ_i is of the form $(\Phi \setminus \{\chi\}) \cup \{\psi_{i,1}, \dots, \psi_{i,k_i}\}$. Now the rule for QZ becomes $\tau : QZ \Rightarrow QZ_1, \dots, QZ_n$ where each Z_i is defined by $\{(\varphi, \varphi) \mid \varphi \in \Phi \setminus \{\chi\}\} \cup \{(\chi, \psi_{i,1}), \dots, (\chi, \psi_{i,k_i})\}$.

Now for every QZ we have fixed a unique rule to be applied, which we will call the applicable rule for QZ . This allows us to define the transition terms of the modal automaton we will construct. Let us first construct the modal automaton \mathbb{A}_ξ before worrying about its acceptance condition.

Definition 4.6. Let ξ be a CTL_Σ^* formula and let $Q_I(\varphi_I)$ be its equivalent Dam term. The modal automaton \mathbb{A}_ξ generated by ξ is $\mathbb{A}_\xi = (A_\xi, \Delta, \text{Acc}, Q_I Z_I)$, where $Z_I := \{(\varphi_I, \varphi_I)\}$, where $A_\xi := \{QZ \mid Q \in \{\mathbf{E}, \mathbf{A}\}, Z \in \mathcal{Z}(\text{Cl}(\varphi_I))\}$, where $\Delta(QZ) = \ell$ if QZ is terminal and $I : QZ \Rightarrow \ell$ is the applicable rule, where $\Delta(QZ) := \tau(Q_1Z_1, \dots, Q_nZ_n)$ if QZ is non-terminal and $\tau : QZ \Rightarrow Q_1Z_1, \dots, Q_nZ_n$ is the applicable rule and where Acc accepts a sequence $Q_0Z_0, Q_1Z_1, Q_2Z_2, \dots$ through \mathbb{A}_ξ if either there is k with $Q_i = \mathbf{E}$ for all $i \geq k$ and all infinite traces through Z_1, Z_2, Z_3, \dots end up in an **R**-cell, or there is k with $Q_i = \mathbf{A}$ for all $i \geq k$ and at least one infinite trace through Z_1, Z_2, Z_3, \dots ends up in an **R**-cell. \triangleleft

The following proposition helps clarify the definition of Acc. It implies that for every infinite sequence Q_0Z_0, Q_1Z_1, \dots through \mathbb{A}_ξ there is a k such that $Q_i = Q_k$ for all $i \geq k$. Now if $Q_k = \mathbf{E}$ then Acc requires all infinite traces to end up in an R-cell, and if $Q_k = \mathbf{A}$ then there needs to be at least one such trace.

Proposition 4.7. *Let C be a cluster of \mathbb{A}_ξ for some ξ . If $QZ, Q'Z' \in C$ then $Q = Q'$ and $\text{Cl}(\text{Ran}(Z)) = \text{Cl}(\text{Ran}(Z'))$.*

Proof. Let $QZ, Q'Z' \in C$, then by definition of a cluster there are $Q_0, \dots, Q_n, \dots, Q_{n+k}, Z_0, \dots, Z_n, \dots, Z_{n+k}$ and $\Phi_0, \dots, \Phi_n, \dots, \Phi_{n+k}$, such that $Q_0 = Q, Q_n = Q', Q_{n+k} = Q, Z_0 = Z, Z_n = Z', Z_{n+k} = Z, \Phi_i = \text{Ran}(Z_i)$ for $i \leq n+k$ and $Q_iZ_i \rightarrow Q_{i+1}Z_{i+1}$ for $i < n+k$. Per construction of \mathbb{A}_ξ we get $\tau_i : Q_i\Phi_i \Rightarrow \dots, Q_{i+1}\Phi_{i+1}, \dots$ for $i < n+k$. By Proposition 4.3 this means that $\text{Cl}(\Phi_0) \subseteq \dots \subseteq \text{Cl}(\Phi_n) \subseteq \dots \subseteq \text{Cl}(\Phi_{n+k}) = \text{Cl}(\Phi_0)$ and therefore $Q_0 = \dots = Q_n = \dots = Q_{n+k} = Q_0$. In particular $Q = Q_0 = Q_n = Q'$ and $\text{Cl}(\text{Ran}(Z)) = \text{Cl}(\Phi_0) = \text{Cl}(\Phi_n) = \text{Cl}(\text{Ran}(Z'))$. \square

With this proposition in mind, we can discern between “E-clusters”, where each state is of the form EZ, and “A-clusters”, where the states are of the form AZ.

Proposition 4.8. *For any CTL_Σ^* formula ξ the automaton \mathbb{A}_ξ has dominated clusters.*

Proof. Let ξ be a CTL_Σ^* formula. For every $QZ \in A_\xi$ we have that $\Delta(QZ)$ is an instance of a Dam rule. The resulting terms are “shallow”, in that they can be generated by

$$\varphi ::= \ell \mid a \mid \diamond a \mid \square a \mid a \vee a \mid a \wedge a$$

where $a \in A$.

Define the quantifier-depth of a CTL^* formula, $\text{qd}(\varphi)$, by

$$\begin{aligned} \text{qd}(\ell) &:= 0 \\ \text{qd}(X\varphi) &:= \text{qd}(\varphi) \\ \text{qd}(\varphi \star \psi) &:= \max\{\text{qd}(\varphi), \text{qd}(\psi)\} & (\star \in \{\vee, \wedge, \mathbf{U}, \mathbf{R}\}) \\ \text{qd}(Q\varphi) &:= \text{qd}(\varphi) + 1 & (Q \in \{\mathbf{E}, \mathbf{A}\}) \end{aligned}$$

and define qd on A_ξ by $\text{qd}(QZ) := \max\{\text{qd}(\varphi) \mid \varphi \in \text{Cl}(\text{Ran}(Z))\}$. By Proposition 4.7 we have $\text{qd}(a) = \text{qd}(b)$ whenever a and b are in the same cluster.

Let C be an E-cluster of \mathbb{A}_ξ . There are no $a \in C$ with $\Delta(a) = \square b$ because there is no E-rules that uses \square . Let $a \in C$ with $\Delta(a) = b_1 \wedge b_2$, then the applicable rule for a was an extraction rule. Looking at these rules gives us that b_2 is either terminal or of the form $Q\{(Q\varphi, \varphi)\}$; in the latter case we find $\text{qd}(b) = \text{qd}(\varphi) < \text{qd}(Q\varphi) \leq \text{qd}(a)$. Either way $b_2 \not\rightsquigarrow a$ hence $b_2 \notin C$. Therefore C is \exists -dominated.

Dually all A-clusters are \forall -dominated. \square

Before we can use duality in the proof of the correctness of this automaton construction, we will need to make the acceptance condition more symmetric. Any sequence through \mathbb{A}_ξ

will eventually stay in some cluster. For \exists -dominated clusters Acc states that all traces must end up in an R-cell in order for \exists to win. For \forall -dominated clusters it states that in order for \forall to win, all traces must not end up in an R-cell. With Proposition 3.7 in mind, this is equivalent to saying that all traces must end up in a U-cell if we can show that none of the traces are stalling. In fact, there is an upper bound on the number of \rightarrow -steps you can take before you reach a state whose applicable rule is modal.

First, let us define a measure of complexity that can capture this upper bound.

Definition 4.9. Define the *immediate complexity* of a CTL* formula, $\text{ic}(\varphi)$, by

$$\begin{aligned} \text{ic}(\ell) &:= 1 \\ \text{ic}(\mathbf{X}\varphi) &:= 0 \\ \text{ic}(\varphi \star \psi) &:= \text{ic}(\varphi) + \text{ic}(\psi) + 1 && (\star \in \{\vee, \wedge, \mathbf{U}, \mathbf{R}\}) \\ \text{ic}(Q\varphi) &:= \text{ic}(\varphi) + 1 && (Q \in \{\mathbf{E}, \mathbf{A}\}) \end{aligned}$$

and define ic on A_ξ by $\text{ic}(QZ) = \sum_{\psi \in \text{Ran}(Z)} \text{ic}(\psi)$. \triangleleft

Analyzing the Dam rules gives us that if ic decreases when applying a rule, unless that rule is a modal rule. Because modal rules are only applied to QZ when all the formulas in the range of Z are of the form $\mathbf{X}\varphi$, this means the traces that go through Z will have to go from $\mathbf{X}\varphi$ to φ . The following proposition formalizes this.

Proposition 4.10. *If Q_1Z_1, Q_2Z_2, \dots is a sequence of states of \mathbb{A}_ξ with $Q_iZ_i \rightarrow Q_{i+1}Z_{i+1}$ for all i , then all infinite traces through Z_1, Z_2, \dots are complete.*

Proof. Let Q_1Z_1, Q_2Z_2, \dots be as such, then there is m such that all Q_iZ_i for $i \geq m$ belong to the same cluster C . By Proposition 4.7 we have $Q_i = Q_m$ for all $i \geq m$. Let ic be the immediate complexity on \mathbb{A}_ξ defined in Definition 4.9, then $\text{ic}(Q_iZ_i) > \text{ic}(Q_{i+1}Z_{i+1})$ unless the rule applied to Q_iZ_i is a modal rule. Of course it is impossible that a decreasing sequence $\text{ic}(Q_iZ_i) > \text{ic}(Q_{i+1}Z_{i+1}) > \dots$ continues forever. This means that for every k there is $i \geq k$ such that Z_i contains only pairs of the form $(\mathbf{X}\psi, \psi)$. Therefore if $\varphi_0, \varphi_1, \dots$ is an infinite trace through Z_1, Z_2, \dots , then $\varphi_{i+1} = \mathbf{X}\varphi_i$ (hence $\varphi_i \neq \varphi_{i+1}$) for infinitely many i 's and thus the trace is non-stalling. \square

We are now ready to show that the automaton \mathbb{A}_ξ is indeed equivalent to ξ .

Lemma 4.11. *Let ξ be a CTL* $_\Sigma$ formula. Let \mathbb{S} be a transition system. If $QZ \in A_\xi$, $\Phi = \text{Ran}(Z)$ and $s \in S$, then $\mathbb{S}, s \Vdash Q\Phi$ iff $\mathbb{S}, s \Vdash (\mathbb{A}_\xi, QZ)$. In particular $\xi \equiv \mathbb{A}_\xi$.*

Proof. We will prove

$$(\forall s \in S) \quad \mathbb{S}, s \Vdash Q\Phi \iff \mathbb{S}, s \Vdash (\mathbb{A}_\xi, QZ) \quad (\star)$$

where $\Phi := \text{Ran}(Z)$ by induction on the cluster-depth of $QZ \in A_\xi$. Once we have this, we use $\xi \equiv Q_I(\varphi_I) \equiv (\mathbb{A}_\xi, Q_I Z_I) \equiv \mathbb{A}_\xi$ to get the particular case.

For the base case we have that the cluster-depth of QZ is 0, which means that there are no $a \in A$ with $QZ \rightarrow a$. It must be that QZ is terminal. Let $I : Q\Phi \Rightarrow \ell$ be the applicable rule, then the only possible match of the game $\mathcal{A}(\mathbb{A}_\xi, \mathbb{S})@(s, QZ)$ goes $(s, QZ) \rightarrow (s, \ell)$, where \exists wins if $\mathbb{S}, s \Vdash \ell$ and loses otherwise. Thus $\mathbb{S}, s \Vdash (\mathbb{A}_\xi, QZ)$ iff $\mathbb{S}, s \Vdash \ell$. By Proposition 4.1 we have $\mathbb{S}, s \Vdash \ell$ iff $\mathbb{S}, s \Vdash Q\Phi$. This concludes the base case.

For the inductive case let $d > 0$ be the cluster-depth of QZ . The induction hypothesis states that (\star) holds for all states in A of cluster-depth less than d . Note that whenever $QZ \rightarrow a$, either a is in the same cluster as QZ or a has cluster-depth less than d . Note that since there is $a \in A$ with $QZ \rightarrow a$, $Q\Phi$ is non-terminal. Let $\tau : QZ \Rightarrow Q_1Z_1, \dots, Q_kZ_k$ be the applicable rule for $\mathbf{E}Z$. Let $\{\varphi_1, \dots, \varphi_l\} = \Phi$.

We will first prove (\star) for states $\mathbf{E}Z$. We have $\mathbb{S}, s \Vdash \mathbf{E}\Phi$ iff $(s, \mathbf{E}\Phi) \in \text{Win}_\exists(\mathcal{E}(\mathbf{E}\Phi, \mathbb{S}))$ iff there is a path π starting at s such that $(\pi, \varphi_1 \wedge \dots \wedge \varphi_l) \in \text{Win}_\exists(\mathcal{E}(\mathbf{E}\Phi, \mathbb{S}))$. By Proposition 3.20 we also have $\mathbb{S}, s \Vdash (\mathbb{A}_\xi, \mathbf{E}Z)$ iff $(s, \mathbf{E}Z) \in \text{Win}_\exists(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}))$ iff there is a path π starting at s such that $(0, \mathbf{E}Z) \in \text{Win}_\exists(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}, C, \pi))$, where C is the cluster $\mathbf{E}Z$ belongs to. We shall show that

$$(\pi, \varphi_1 \wedge \dots \wedge \varphi_l) \in \text{Win}_\exists(\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})) \iff (0, \mathbf{E}Z) \in \text{Win}_\exists(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}, C, \pi))$$

holds for every path π through \mathbb{S} .

(\implies)

Suppose that f is a winning strategy of \exists for $\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})@(\pi, \varphi_1 \wedge \dots \wedge \varphi_l)$. By Proposition 3.14 we may assume that f is positional. Some matches starting there continue through choices of \forall to (π, φ_i) ; these partial matches are consistent with f since \exists hasn't made any choices yet. This means that each (π, φ) for $\varphi \in \Phi$ is a winning position for \exists under f . While \exists and \forall play a match of the cluster-path acceptance game $\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}, C, \pi)@(0, \mathbf{E}Z)$, \exists creates a list $\mathbf{E}Z, \mathbf{E}Z_1^*, \mathbf{E}Z_2^*, \dots$ of the states in C that are visited. This list is such that if the acceptance match is at $(j, \mathbf{E}Z_r^*)$ and $\mathbf{E}Z_r^* \in C$, then the list is $\mathbf{E}Z, \mathbf{E}Z_1^*, \dots, \mathbf{E}Z_r^*$, every position (π^j, ψ) for $\psi \in \text{Ran}(Z_r^*)$ is a winning position for \exists in the evaluation game, and for every trace through Z_1^*, \dots, Z_r^* there is an f -consistent match of the evaluation game that goes through the formulas in the trace in order. The real match starts at $(0, \mathbf{E}Z)$, the first visited state is $\mathbf{E}Z$ and each (π, φ) for $\varphi \in \Phi = \text{Ran}(Z)$ is indeed winning for \exists .

When the game is at a position $(j, \mathbf{E}Z_r^*)$ with $\mathbf{E}Z_r^* \in C$, it necessarily continues to $(j, \Delta(\mathbf{E}Z_r^*))$. How the acceptance match continues from there depends on the applicable rule for $\mathbf{E}\Psi$ where $\Psi = \text{Ran}(Z_r^*)$, which is some $\tau : \mathbf{E}\Psi \Rightarrow Q_1\Psi_1, \dots, Q_k\Psi_k$. Note that if $\Delta(\mathbf{E}Z_r^*) = \tau(Q_1Z_1, \dots, Q_kZ_k)$ and $\text{Ran}(Z_i) = \Psi_i$ for each i . Let $\{\psi_1, \dots, \psi_n\} = \Psi$.

If the applicable rule is a modal rule then we have $\tau = \diamond$, $k = 1$, each ψ_i is of the form $\mathbf{X}\psi'_i$, $Q_1 = \mathbf{E}$, $\Psi_1 = \{\psi'_1, \dots, \psi'_n\}$ and $Z_1 = \{(\mathbf{X}\psi'_1, \psi'_1), \dots, (\mathbf{X}\psi'_n, \psi'_n)\}$. Now the acceptance match is at $(j, \diamond(\mathbf{E}Z_1))$. All evaluation matches that go through $(\pi^j, \mathbf{X}\psi'_i)$ necessarily continue to (π^{j+1}, ψ'_i) ; since each former position is winning for \exists , so must each latter be winning for \exists . If $\mathbf{E}Z_1 \in C$ then the acceptance match necessarily continues to $(j+1, \mathbf{E}Z_1)$. From there the acceptance match moves on with $\mathbf{E}Z_{r+1} = \mathbf{E}Z_1$ as the last visited state. An evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in $(\pi^j, \mathbf{X}\psi'_i)$ can be

extended with (π^{j+1}, ψ'_i) . If $\mathbf{E}Z_1 \notin C$ then \exists has to choose a move $(t, \mathbf{E}\Psi_1)$ where $\pi(j)Rt$. She chooses $(\pi(j+1), \mathbf{E}\Psi_1)$. After this move, the acceptance match ends.

If the applicable rule is not modal, then it affects exactly one of the ψ_i . Let e be the index of the affected element of Ψ . What exactly the applicable rule is now only depends on the shape of ψ_e , which is one of ℓ , $\mathbf{E}\Theta$, $\mathbf{A}\Theta$, $\vartheta_1 \wedge \vartheta_2$, $\vartheta_1 \vee \vartheta_2$, $\vartheta_1 \mathbf{U} \vartheta_2$ and $\vartheta_1 \mathbf{R} \vartheta_2$. Some Ψ_i will contain $\Psi \setminus \{\psi_e\}$ together with some other formulas; note that every (π^j, ψ_i) will remain a winning position thus it is enough to check these extra formulas. Similarly an evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in (π^j, ψ) with $\psi \neq \psi_e$ is also an evaluation match of the trace that goes through Z_1^*, \dots, Z_r^*, Z_i if $(\psi, \psi) \in Z_i$.

If $\psi_e = \mathbf{E}\Theta$ or $\psi_e = \mathbf{A}\Theta$, and $\Psi = \{\psi_e\}$, then $\tau = I$, $k = 1$, $\text{Ran}(Z_1) = \Theta$ and $QZ \rightarrow Q_1Z_1$. The acceptance match goes from $(j, \mathbf{E}Z_r^*)$ to (j, Q_1Z_1) . Now $Q_1Z_1 \notin C$ and therefore the match continues to $(\pi(j), Q_1Z_1)$, where it ends.

If $\psi_e = \ell$, $\psi_e = \mathbf{E}\Theta$ or $\psi_e = \mathbf{A}\Theta$, and $\Psi \setminus \{\psi_e\} \neq \emptyset$, then $\tau = \wedge$, $k = 2$, $Q_1 = \mathbf{E}$, $\Psi_1 = \Psi \setminus \{\psi_e\}$, $Z_1 = \{(\psi, \psi) \mid \psi \in \Psi_1\}$, $Q_2\Phi_2 \equiv \psi_e$ and $QZ \rightarrow Q_2Z_2$. The acceptance match goes from $(j, \mathbf{E}Z_r^*)$ to $(j, \mathbf{E}Z_1 \wedge Q_2Z_2)$, where \forall has to choose one of $(j, \mathbf{E}Z_1)$ and (j, Q_2Z_2) . If he chooses the latter then $Q_2Z_2 \notin C$ and therefore the match continues to $(\pi(j), Q_2Z_2)$, where it ends. If he chooses the former then either $\mathbf{E}Z_1 \notin C$ and the match ends at $(\pi(j), \mathbf{E}Z_1)$, or $\mathbf{E}Z_1 \in C$ and the match necessarily continues to $(j, \Delta(\mathbf{E}\Psi_1))$. In that case the acceptance match moves on with $\mathbf{E}Z_{r+1}^* = \mathbf{E}Z_1$ as the last visited state. An evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in (π^j, ψ_i) with $\psi_i \neq \psi_e$ is also an evaluation match for the trace that goes through Z_1^*, \dots, Z_r^*, Z_1 .

If $\psi_e = \vartheta_1 \wedge \vartheta_2$ then we have $\tau = I$, $k = 1$, $Q_1 = \mathbf{E}$, $\Psi_1 = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_1, \vartheta_2\}$ and $Z_1 = \{(\psi_e, \vartheta_1), (\psi_e, \vartheta_2)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$. Some of the evaluation matches going through (π^j, ψ_e) continue to (π^j, ϑ_1) and others to (π^j, ϑ_2) . Since this is a choice of \forall both are consistent with f , hence both (π^j, ϑ_1) and (π^j, ϑ_2) are winning positions for \exists . The acceptance match necessarily continues to $(j, \mathbf{E}Z_1)$ and moves on with $\mathbf{E}Z_{r+1}^* = \mathbf{E}Z_1$ as the last visited state. An evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in (π^j, ψ_e) can be extended with each (ψ_e, ϑ_i) since any choice of \forall is consistent with f .

If $\psi_e = \vartheta_1 \vee \vartheta_2$ then we have $\tau = \vee$, $k = 2$ and $Q_i = \mathbf{E}$, $\Psi_i = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_i\}$, $Z_i = \{(\psi_e, \vartheta_i)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$ for both i . Evaluation matches consistent with f that go through (π^j, ψ_e) continue to either (π^j, ϑ_1) or (π^j, ϑ_2) as instructed by f . The acceptance match is at the position $(j, Q_1Z_1 \vee Q_2Z_2)$ and there \exists must choose between (j, Q_1Z_1) and (j, Q_2Z_2) . If the f -consistent evaluation matches continue to (π^j, ϑ_i) then she chooses (j, Q_iZ_i) ; note that in that case (π^j, ϑ_i) is a winning position for \exists . Now either $Q_iZ_i \notin C$ and the match ends at $(\pi(j), Q_iZ_i)$, or $Q_iZ_i \in C$ and the match necessarily continues to $(j, \Delta(Q_iZ_i))$. In that case the acceptance match moves on with $\mathbf{E}Z_{r+1}^* = Q_1Z_i$ as the last visited state. An evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in (π^j, ψ_e) can be extended with (π^j, ϑ_i) , since this is consistent with f , to become an evaluation match for the trace through Z_1^*, \dots, Z_r^*, Z_i .

If $\psi_e = \vartheta_1 \mathbf{U} \vartheta_2$ then we have that $\tau = \vee$, $k = 2$, $Q_1 = \mathbf{E}$, $\Psi_1 = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_2\}$, $Z_1 = \{(\psi_e, \vartheta_2)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$, $Q_2 = \mathbf{E}$, $\Psi_2 = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_1, \mathbf{X}\psi_e\}$ and $Z_2 = \{(\psi_e, \vartheta_1), (\psi_e, \mathbf{X}\psi_e)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$. The evaluation matches that go through (π^j, ψ_e) necessarily continue to $(\pi^j, \vartheta_2 \vee (\vartheta_1 \wedge \mathbf{X}\psi_e))$, and then continue to either

(π^j, ϑ_2) or $(\pi^j, \vartheta_1 \wedge \mathbf{X}\psi_e)$ as instructed by f . This again helps \exists choose between (j, Q_1Z_1) and (j, Q_2Z_2) . Note that in the case that \exists chooses (j, Q_2Z_2) , some evaluation matches that go through $(\pi^j, \vartheta_1 \wedge \mathbf{X}\psi_e)$ continue via an \forall -move to (π, ϑ_1) and others to $(\pi, \mathbf{X}\psi_e)$. Therefore either (π^j, ϑ_2) is a winning position or both (π^j, ϑ_1) and $(\pi^j, \mathbf{X}\psi_e)$ are. Again the acceptance match either ends at $(\pi(j), Q_iZ_i)$ or moves on with Q_iZ_i as the last visited state. An evaluation match that goes through a trace of Z_1^*, \dots, Z_r^* and ends in (π^j, ψ_e) can be extended with (π^j, ϑ_2) , with $(\pi^j, \vartheta_1 \wedge \mathbf{X}\psi_e)$ and (π^j, ϑ_1) or with $(\pi^j, \vartheta_1 \wedge \mathbf{X}\psi_e)$ and $(\pi^j, \mathbf{X}\psi_e)$, whichever is appropriate.

If $\psi_e = \vartheta_1\mathbf{R}\vartheta_2$ then we have that $\tau = \vee$, $k = 2$, $Q_1 = \mathbf{E}$, $\Psi_1 = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_2, \vartheta_1\}$, $Z_1 = \{(\psi_e, \vartheta_2), (\psi_e, \vartheta_1)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$, $Q_2 = \mathbf{E}$, $Ps_2 = (\Psi \setminus \{\psi_e\}) \cup \{\vartheta_2, \mathbf{X}\psi_e\}$ and $Z_2 = \{(\psi_e, \vartheta_2), (\psi_e, \mathbf{X}\psi_e)\} \cup \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$. The evaluation matches that go through (π^j, ψ_e) necessarily continue to $(\pi^j, \vartheta_2 \wedge (\vartheta_1 \vee \mathbf{X}\psi_e))$ and from there some continue via an \forall -move to (π^j, ϑ_2) and others to $(\pi^j, \vartheta_1 \vee \mathbf{X}\psi_e)$. Both (π^j, ϑ_2) and $(\pi^j, \vartheta_1 \vee \mathbf{X}\psi_e)$ must be winning positions for \exists . For the evaluation matches that go through $(\pi^j, \vartheta_1 \vee \mathbf{X}\psi_e)$, the strategy f instructs \exists to continue either to (π^j, ϑ_1) or to $(\pi^j, \mathbf{X}\psi_e)$. This helps \exists choose between (j, Q_1Z_1) and (j, Q_2Z_2) ; note that the chosen position in the evaluation match is a winning position for \exists . Again the acceptance match either ends at $(\pi(j), Q_iZ_i)$ or moves on with Q_iZ_i as the last visited state. The evaluation matches that go through traces of Z_1^*, \dots, Z_r^* can again be extended with (π^j, ϑ_2) , with $(\pi^j, \vartheta_1 \vee \mathbf{X}\psi_e)$ and (π^j, ϑ_1) or with $(\pi^j, \vartheta_1 \vee \mathbf{X}\psi_e)$ and $(\pi^j, \mathbf{X}\psi_e)$, whichever is appropriate.

In this manner \exists and \forall play an acceptance match, with \exists keeping track of the visited states $\mathbf{E}Z, \mathbf{E}Z_1^*, \mathbf{E}Z_2^*, \dots$.

Suppose the acceptance match ends after finitely many moves at (π^j, QZ) with $QZ \notin C$. Let $\Psi = \text{Ran}(Z)$. If this happened after a point where $\psi_e = \ell$, $\psi_e = \mathbf{E}\Theta$ or $\psi_e = \mathbf{A}\Theta$, then we have $Q\Psi \equiv \psi_e$. Now per construction (π^j, ψ_e) is a winning position for \exists in the game $\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})$, therefore $\mathbb{S}, \pi^j \Vdash \psi_e$, hence $\mathbb{S}, \pi^j \Vdash Q\Psi$ and thus $\mathbb{S}, \pi(j) \Vdash Q\Psi$. If not then $Q = \mathbf{E}$; let $\{\psi_1, \dots, \psi_n\} = \Psi$. Per construction each of the positions (π^j, ψ_i) is a winning position for \exists in the game $\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})$. By Proposition 3.12 they are also winning positions in the game $\mathcal{E}(\mathbf{E}\Psi, \mathbb{S})$. This means $(\pi^j, \psi_1 \wedge \dots \wedge \psi_n)$ is a winning position for \exists in that game, and therefore so is $(\pi(j), \mathbf{E}\Psi)$. Hence $\mathbb{S}, s \Vdash \mathbf{E}\Psi$. Either way we find $\mathbb{S}, \pi(j) \Vdash \mathbf{E}\Psi$. Since $\mathbf{E}Z \notin C$ we can use the induction hypothesis to get $(\pi(j), \mathbf{E}Z) \in \text{Win}_{\exists}(\mathcal{A}(\mathbb{A}_{\xi}, \mathbb{S}))$. Therefore \exists wins the acceptance match.

Suppose the acceptance match is infinite, then the winner of this match depends on the induced sequence of states, all of which are in C . Let $\mathbf{E}Z, \mathbf{E}Z_1, \mathbf{E}Z_2, \dots$ be this sequence and let $\varphi_0, \varphi_1, \varphi_2, \dots$ be a trace through Z_1, Z_2, \dots , then by Proposition 4.10 this trace is non-stalling. Per construction there is a match of the evaluation game that is consistent with f and that goes through the formulas in the trace in order. The induced trace of that match and the trace $\varphi_0, \varphi_1, \dots$ are therefore mutually cofinal. By Proposition 3.9 and Proposition 3.7, both traces end up in the same cell, which is either a U-cell or an R-cell. Because f is a winning strategy the evaluation match is won by \exists , and thus the cell in question is an R-cell. We conclude that all traces through Z_1, Z_2, \dots end up in an R-cell and therefore \exists wins the acceptance match.

(\Leftarrow)

Suppose that g is a winning strategy of \exists for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi) @ (0, \mathbf{E}Z)$. If \exists and \forall play a match of $\mathcal{E}(\mathbf{E}\Phi, \mathbb{S}) @ (\pi, \varphi_1 \wedge \dots \wedge \varphi_l)$, then this match continues through a series of \forall -moves to some (π, φ_i) . During the match \exists plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi) @ (0, \mathbf{E}Z)$ that is consistent with g , creates a sequence of continuations $Z_0^*, Z_1^*, Z_2^*, \dots$, and creates a trace $\varphi_0^*, \varphi_1^*, \varphi_2^*, \dots$ through Z_1^*, Z_2^*, \dots such that whenever the real match is at (π^j, φ_k^*) the shadow match is at $(j, \mathbf{E}Z_k^*)$ and $\varphi_k^* \in \text{Ran}(Z_k^*)$. To start \exists defines $\varphi_0^* := \varphi_i$ and $Z_0^* := Z$. Note $\varphi_i \in \text{Ran}(Z)$. The acceptance match indeed begins at $(0, \mathbf{E}Z_0^*)$.

Suppose the real match is at (π^j, φ_k^*) and the acceptance match is at $(j, \mathbf{E}Z_k^*)$. How the matches continue depends on the Dam rule that is applicable to $\mathbf{E}\Psi$ where $\Psi = \text{Ran}(Z_k^*)$.

If it is the modal rule, then every $\psi \in \Psi$ is of the form $\mathbf{X}\psi'$. In particular $\varphi_k^* = \mathbf{X}\varphi'$. The evaluation match necessarily continues to (π^{j+1}, φ') and the shadow match necessarily continues to $(j+1, \mathbf{E}Z')$ where $Z' = \{(\mathbf{X}\psi', \psi') \mid \mathbf{X}\psi' \in \Psi\}$. Note that because $(\varphi_k^*, \varphi') \in Z'$ we have that $\varphi_0^*, \varphi_1^*, \dots, \varphi_k^*, \varphi'$ is a trace through $Z_1^*, Z_2^*, \dots, Z_k^*, Z'$ and that $\varphi' \in \text{Ran}(Z')$. Thus we define $\varphi_{k+1}^* = \varphi'$ and $Z_{k+1}^* = Z'$.

If it is not a modal rule, then it affects exactly one of the $\psi \in \Psi$, say ψ_e . What exactly the applicable rule is now only depends on the shape of ψ_e , which is one of ℓ , $\mathbf{E}\Theta$, $\mathbf{A}\Theta$, $\vartheta_1 \wedge \vartheta_2$, $\vartheta_1 \vee \vartheta_2$, $\vartheta_1 \mathbf{U} \vartheta_2$ and $\vartheta_1 \mathbf{R} \vartheta_2$. Define $Z' := \{(\psi, \psi) \mid \psi \in \Psi \setminus \{\psi_e\}\}$

If $\psi_e = \ell$, $\psi_e = \mathbf{E}\Theta$ or $\psi_e = \mathbf{A}\Theta$, and $\psi_e \neq \varphi_k^*$, then the acceptance match necessarily continues to $(j, \mathbf{E}Z' \wedge Q_2 Z_2)$ and $(\varphi_k^*, \varphi_k^*) \in Z'$. Here shadow- \forall chooses $(j, \mathbf{E}Z')$, and we define $\varphi_{k+1}^* = \varphi_k^*$ and $Z_{k+1}^* = Z'$.

If $\psi_e = \ell$ and $\psi_e = \varphi_k^*$, then $\mathbf{E}Z_k^*$ is terminal. Now the acceptance match necessarily continues to (j, ℓ) where the game ends, with \exists winning if $\mathbb{S}, \pi(j) \Vdash \ell$ and losing otherwise. The real match also ends here, with \exists winning in exactly the same conditions.

If $\psi_e = \mathbf{E}\Theta$ or $\psi_e = \mathbf{A}\Theta$ and $\psi_e = \varphi_k^*$, then there is $Q_i Z_i$ with $\mathbf{E}Z_k^* \rightarrow Q_i Z_i$ and $Q_i Z_i \equiv \varphi_k^*$. Now τ is either I or \wedge ; in the latter case shadow- \forall chooses $(j, Q_i Z_i)$ and in the former case this is automatic. From there the acceptance match necessarily continues to $(\pi(j), Q_i Z_i)$ since $Q_i Z_i \notin C$, and there it ends. Now because this match is consistent with g , it is won by \exists . This means $\mathbb{S}, \pi(j) \Vdash (\mathbb{A}, Q_i Z_i)$. Because $Q_i Z_i \notin C$ we can use the induction hypothesis to get $\mathbb{S}, \pi(j) \Vdash Q_i Z_i$, and therefore $\mathbb{S}, \pi(j) \Vdash \varphi_k^*$. This in turn means that $(\pi(j), \varphi_k^*)$ is a winning position in $\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})$. The real match necessarily continues from (π^j, φ_k^*) to $(\pi(j), \varphi_k^*)$, hence is eventually won by \exists .

If ψ_e is of the form $\vartheta_1 \wedge \vartheta_2$, $\vartheta_1 \vee \vartheta_2$, $\vartheta_1 \mathbf{U} \vartheta_2$ or $\vartheta_1 \mathbf{R} \vartheta_2$, and $\psi_e \neq \varphi_k^*$, then whichever $Q_i Z_i$ is chosen (in accordance with g) will have $(\varphi_k^*, \varphi_k^*) \in Z' \subseteq Z_i$. We define $\varphi_{k+1}^* = \varphi_k^*$ and $Z_{k+1}^* = Z_i$.

If $\varphi_k^* = \psi_e = \vartheta_1 \wedge \vartheta_2$, then in the evaluation match \forall can choose either (π^j, ϑ_1) or (π^j, ϑ_2) . Say he chooses (π^j, ϑ_i) . The acceptance match necessarily continues to $(j, \mathbf{E}Z_1)$ where $Z_1 = \{(\varphi_k^*, \vartheta_1), (\varphi_k^*, \vartheta_2)\} \cup Z'$. We define $\varphi_{k+1}^* = \vartheta_i$ and $Z_{k+1}^* = Z_1$.

If $\varphi_k^* = \psi_e = \vartheta_1 \vee \vartheta_2$, then the acceptance match continues to $(j, \mathbf{E}Z_1 \vee \mathbf{E}Z_2)$ where $Z_i = \{(\varphi_k^*, \vartheta_i)\} \cup Z'$ for each i . Here g instructs \exists to choose either $(j, \mathbf{E}Z_1)$ or $(j, \mathbf{E}Z_2)$. This helps \exists choose between (π^j, ϑ_1) and (π^j, ϑ_2) . If shadow- \exists is instructed to choose $(j, \mathbf{E}Z_i)$ then \exists chooses (π^j, ϑ_i) and defines $\varphi_{k+1}^* = \vartheta_i$ and $Z_{k+1}^* = Z_1$.

If $\varphi_k^* = \psi_e = \vartheta_1 \mathbf{U} \vartheta_2$, then the acceptance match continues to $(j, \mathbf{E}Z_1 \vee \mathbf{E}Z_2)$ where

$Z_1 = \{(\varphi_k^*, \vartheta_2)\} \cup Z'$ and $Z_2 = \{(\varphi_k^*, \vartheta_1), (\varphi_k^*, \mathbf{X}\varphi_k^*)\} \cup Z'$. Here g instructs \exists to choose either $(j, \mathbf{E}Z_1)$ or $(j, \mathbf{E}Z_2)$. The evaluation match necessarily continues from (π^j, φ_k^*) to $(\pi^j, \vartheta_2 \vee (\vartheta_1 \wedge \mathbf{X}\varphi_k^*))$. Here \exists chooses (π^j, ϑ_2) if g instructs shadow- \exists to pick $(j, \mathbf{E}Z_1)$, and $(\pi^j, \vartheta_1 \wedge \mathbf{X}\varphi_k^*)$ if g instructs shadow- \exists to pick $(j, \mathbf{E}Z_2)$. In the latter case, \forall chooses either (π^j, ϑ_1) or $(\pi^j, \mathbf{X}\varphi_k^*)$. In this way the acceptance match is at a position $(j, \mathbf{E}Z_i)$ and the evaluation match is at a position (π^j, φ') where $\varphi' \in \{\vartheta_1, \vartheta_2, \mathbf{X}\varphi_k^*\}$. Note $(\varphi_k^*, \varphi') \in Z_i$. Define $\varphi_{k+1}^* = \varphi'$ and $Z_{k+1}^* = Z_i$.

If $\varphi_k^* = \psi_e = \vartheta_1 \mathbf{R}\vartheta_2$, then the acceptance match continues to $(j, \mathbf{E}Z_1 \vee \mathbf{E}Z_2)$ where $Z_1 = \{(\varphi_k^*, \vartheta_2), (\varphi_k^*, \vartheta_1)\} \cup Z'$ and $Z_2 = \{(\varphi_k^*, \vartheta_2), (\varphi_k^*, \mathbf{X}\varphi_k^*)\} \cup Z'$. Here g instructs \exists to choose either $(j, \mathbf{E}Z_1)$ or $(j, \mathbf{E}Z_2)$. The evaluation match necessarily continues from (π^j, φ_k^*) to $(\pi^j, \vartheta_2 \wedge (\vartheta_1 \vee \mathbf{X}\varphi_k^*))$. If \forall chooses $\vartheta_1 \vee \mathbf{X}\varphi_k^*$ then \exists chooses (π^j, ϑ_1) if g instructs shadow- \exists to pick $(j, \mathbf{E}Z_1)$, and $(\pi^j, \mathbf{X}\varphi_k^*)$ if g instructs shadow- \exists to pick $(j, \mathbf{E}Z_2)$. In this way the acceptance match is at a position $(j, \mathbf{E}Z_i)$ and the evaluation match is at a position (π^j, φ') where $\varphi' \in \{\vartheta_1, \vartheta_2, \mathbf{X}\varphi_k^*\}$. Note $(\varphi_k^*, \varphi') \in Z_i$. Define $\varphi_{k+1}^* = \varphi'$ and $Z_{k+1}^* = Z_i$.

In this way the evaluation match can be played. If the acceptance match is finite then it ends when $\mathbf{E}Z_k^*$ is either terminal or not in C ; in those cases we have seen that \exists wins. If the acceptance match is infinite then its k -th position is of the form $(j, \mathbf{E}Z_k^*)$, and every position in the evaluation match is of the form (π^j, φ) . Furthermore $\varphi_0^*, \varphi_1^*, \varphi_2^*, \dots$ is a trace through Z_1^*, Z_2^*, \dots , and the evaluation match goes through all of the positions (π^j, φ_k^*) in order. By Proposition 4.10 the trace is non-stalling, and hence ends up in either a U-cell or a R-cell. By Proposition 3.9 the evaluation match also ends up in such a cell, and these two cells must be the same. Because the acceptance match is won by \exists all traces through Z_1^*, Z_2^*, \dots end up in an R-cell. That means that the evaluation match ends up in an R-cell, and hence is won by \exists .

We now return to the dual case, i.e. where $Q = \mathbf{A}$. Note that (\star) and

$$(\forall s \in S) \quad \mathbb{S}, s \not\llcorner Q\Phi \iff \mathbb{S}, s \not\llcorner (\mathbb{A}_\xi, QZ)$$

are equivalent. By Proposition 3.14 we have that $\mathbb{S}, s \not\llcorner \mathbf{A}\Phi$ iff $(s, \mathbf{A}\Phi) \in \text{Win}_\forall(\mathcal{E}(\mathbf{A}\Phi, \mathbb{S}))$ iff there is a path π starting at s with $(\pi, \varphi_1 \vee \dots \vee \varphi_l) \in \text{Win}_\forall(\mathcal{E}(\mathbf{A}\Phi, \mathbb{S}))$. By Proposition 2.42 and Proposition 3.20 we also have $\mathbb{S}, s \not\llcorner (\mathbb{A}_\xi, \mathbf{A}Z)$ iff $(s, \mathbf{A}Z) \in \text{Win}_\forall(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}))$ iff there is a path π starting at s such that $(0, \mathbf{A}Z) \in \text{Win}_\forall(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}, C, \pi))$, where C is the cluster $\mathbf{A}Z$ belongs to. It is therefore enough to show that

$$(\pi, \varphi_1 \vee \dots \vee \varphi_l) \in \text{Win}_\forall(\mathcal{E}(\mathbf{E}\Phi, \mathbb{S})) \iff (0, \mathbf{A}Z) \in \text{Win}_\forall(\mathcal{A}(\mathbb{A}_\xi, \mathbb{S}, C, \pi))$$

holds for every path π through \mathbb{S} . The proofs for (\implies) and (\impliedby) are dual to the above. Note in particular that by Proposition 3.7 and Proposition 4.10 any infinite trace through a sequence of states of \mathbb{A}_ξ that doesn't end up in an R-cell must end up in a U-cell. \square

4.2 Automaton modification

In this section we will construct a guarded modal parity automaton that is equivalent to \mathbb{A}_ξ . First we focus on turning the acceptance condition Acc into a parity condition. If we view sequences through \mathbb{A}_ξ as streams over the alphabet A_ξ , then Acc denotes a language over A_ξ . Our strategy is to first find an A_ξ -stream automaton \mathbb{B}_ξ that recognizes Acc and then combine it with \mathbb{A}_ξ to get an automaton that runs both \mathbb{A}_ξ and \mathbb{B}_ξ simultaneously.

For \exists to win a match that stays in an \exists -typical cluster, all traces through the match need to end up in an R-cell. With Proposition 3.6 in mind, she loses if there is even one trace through the match that ends up in a U-cell. If we look at the Dam rule for $\text{E}(\varphi\text{U}\psi)$, we see that only one of the two resulting Dam terms contains $\text{X}(\varphi\text{U}\psi)$. We can see the left term as “resolving” the U-formula, and the right term as “postponing”. Now a sequence through \mathbb{A}_ξ has a trace that ends up in a U-cell whenever there is a U-formula that is postponed infinitely often. Conversely, in order for \exists to win she needs to show that every occurring U-formula is eventually resolved.

The idea of the stream automaton \mathbb{B}_ξ is this: while in an \exists -dominated cluster, \exists keeps a “to-do list” of U-formulas that still need to be resolved. Whenever a U-formula is resolved, it is crossed off the list. Whenever the list becomes empty, a new list of U-formulas is created. Now \exists wins if her to-do list is empty infinitely often. Therefore we assign a priority of 2 to empty list and a priority 1 to non-empty lists. For \forall -dominated clusters, \forall keeps a to-do list of R-formulas; here the empty list has priority 1 and non-empty lists have priority 0. Whenever a match changes from an \exists -dominated cluster to an \forall -dominated cluster or vice versa, the list is discarded and a new one is created.

Definition 4.12. Let ξ be a CTL_Σ^* formula, let $Q_I(\varphi_I)$ be its equivalent Dam term and let \mathbb{A}_ξ be its automaton. Let Φ_U be the set of all U-formulas in $\text{Cl}(\varphi_I)$, and Φ_R the set of all R-formulas in $\text{Cl}(\varphi_I)$. The deterministic parity A_ξ -stream automaton \mathbb{B}_ξ generated by ξ is $\mathbb{B}_\xi := (B_\xi, \delta, \Omega, Q_I\emptyset)$ where $B_\xi := \{\text{E}\Psi \mid \Psi \subseteq \Phi_U\} \cup \{\text{A}\Psi \mid \Psi \subseteq \Phi_R\}$, where $\delta : B_\xi \times A_\xi \rightarrow B_\xi$ is given by

$$\begin{aligned} \delta(\text{E}\emptyset, \text{EZ}) &:= \text{E}\{\varphi \in \Phi_U \mid (\text{X}\varphi, \varphi) \in Z\} \\ \delta(\text{E}\Psi, \text{EZ}) &:= \text{E}\{\varphi \in \Psi \mid \{(\varphi, \varphi), (\text{X}\varphi, \varphi), (\varphi, \text{X}\varphi), (\text{X}\varphi, \text{X}\varphi)\} \cap Z \neq \emptyset\} \\ \delta(\text{E}\Psi, \text{AZ}) &:= \delta(\text{A}\emptyset, \text{AZ}) \\ \delta(\text{A}\Psi, \text{EZ}) &:= \delta(\text{E}\emptyset, \text{EZ}) \\ \delta(\text{A}\emptyset, \text{AZ}) &:= \text{A}\{\varphi \in \Phi_R \mid (\text{X}\varphi, \varphi) \in Z\} \\ \delta(\text{A}\Psi, \text{AZ}) &:= \text{A}\{\varphi \in \Psi \mid \{(\varphi, \varphi), (\text{X}\varphi, \varphi), (\varphi, \text{X}\varphi), (\text{X}\varphi, \text{X}\varphi)\} \cap Z \neq \emptyset\} \end{aligned}$$

and where $\Omega : B_\xi \rightarrow \mathbb{N}$ is given by $\Omega(\text{E}\Psi) := 2$ if $\Psi = \emptyset$, $\Omega(\text{E}\Psi) := 1$ otherwise, $\Omega(\text{A}\Psi) := 1$ if $\Psi = \emptyset$, and $\Omega(\text{A}\Psi) := 0$ otherwise. \triangleleft

For reasons which will become apparent later, a new list is only created directly after a modal step. This is why the transition function for empty lists only adds formulas φ for which $(\text{X}\varphi, \varphi)$ belongs to the continuation. For now, we need to prove that this stream

automaton actually emulates Acc. First, note that every formula in the range of a reachable state in \mathbb{A}_ξ is the end-point of some finite trace.

Proposition 4.13. *If Q_1Z_1, \dots, Q_nZ_n is a sequence of states of \mathbb{A}_ξ with $Q_iZ_i \rightarrow Q_{i+1}Z_{i+1}$ for all i and $\varphi \in \text{Ran}(Z_n)$, then there is a trace $\varphi_0, \dots, \varphi_n$ through Z_1, \dots, Z_n with $\varphi_n = \varphi$.*

Proof. By induction on n . For the base case let $\varphi \in \text{Ran}(Z_1)$, then $(\psi, \varphi) \in Z_1$ for some ψ . Now take $\varphi_0 := \psi$ and $\varphi_1 := \varphi$. For the inductive case, let $n \geq 2$ and suppose it holds for all formulas in $\text{Ran}(Z_{n-1})$. Let $\varphi \in \text{Ran}(Z_n)$, then $(\psi, \varphi) \in Z_n$ for some ψ . If we have a trace $\varphi_0, \dots, \varphi_{n-1}$ through Z_1, \dots, Z_{n-1} with $\varphi_{n-1} = \psi$, then we take $\varphi_n := \varphi$. Thus we are left to prove that $\psi \in \text{Ran}(Z_{n-1})$. In fact per construction $\text{Dom}(Z') \subseteq \text{Ran}(Z)$ whenever $QZ \rightarrow Q'Z'$ in \mathbb{A}_ξ . \square

This allows us to construct a trace that ends up in a U-cell or an R-cell as appropriate whenever we have a to-do list that remains non-empty forever.

Proposition 4.14. *Let ξ be a CTL_Σ^* formula and let \mathbb{A}_ξ be its automaton. The automaton \mathbb{B}_ξ accepts a sequence Q_1Z_1, Q_2Z_2, \dots of \mathbb{A}_ξ if and only if Acc does.*

Proof. Let Q_1Z_1, Q_2Z_2, \dots be a sequence through \mathbb{A}_ξ , then there is m such that all Q_iZ_i for $i \geq m$ belong to the same cluster. By Proposition 4.7 we have $Q_i = Q_m$ for all $i \geq m$. Now we have that Acc accepts the sequence iff Acc accepts $Q_mZ_m, Q_{m+1}Z_{m+1}, \dots$, and the same thing for \mathbb{B}_ξ . Thus we may assume without loss of generality that we have a Q with $Q_i = Q$ for all $i \geq 0$. Note that if b_0, b_1, \dots is the run of \mathbb{B}_ξ on such a sequence, then each b_i for $i \geq 1$ is of the form $Q\Psi_i$ for some Ψ_i .

Suppose $Q = \text{E}$ and Acc rejects the sequence. This means that there is a trace $\varphi_0, \varphi_1, \varphi_2, \dots$ that ends up in a U-cell; let $k \geq 1$ and ψ be such that ψ is a U-formula and $\varphi_i \in \{\psi, \text{X}\psi\}$ for all $i \geq k$. Let $Q_1\emptyset, \text{E}\Psi_1, \text{E}\Psi_2, \dots$ be the run of \mathbb{B}_ξ on the sequence. If $\Psi_i \neq \emptyset$ for all $i \geq k$, then $\Omega(\text{E}\Psi_i) = 1$ for all $i \geq k$ and therefore \mathbb{B}_ξ rejects the sequence. So assume that there is $j \geq k$ such that $\Psi_j = \emptyset$. If $\Psi_i = \emptyset$ then either the rule for $\text{E}\Psi_i$ is not modal and $\Psi_{i+1} = \emptyset$, or $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ and $(\varphi_i, \varphi_{i+1}) = (\text{X}\psi, \psi)$, hence $\psi \in \Psi_{i+1}$. Since the trace is non-stalling, there is $j' > j$ with $\psi \in \Psi_{j'}$. If $i \geq j'$ and $\psi \in \Psi_i$, then from $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ we get that $\psi \in \Psi_{i+1}$. We conclude that $\psi \in \Psi_i$ for all $i \geq j'$, and therefore $\Omega(\text{E}\Psi_i) = 1$ for all $i \geq j'$. This means that \mathbb{B}_ξ rejects the sequence.

Suppose $Q = \text{E}$ and \mathbb{B}_ξ rejects the sequence. Let $Q_1\emptyset, \text{E}\Psi_1, \text{E}\Psi_2, \dots$ be the run of \mathbb{B}_ξ on the sequence, then there is $k \geq 1$ with $\Psi_i \neq \emptyset$ for all $i \geq k$. Observe that $\Psi' \subseteq \Psi$ if $\text{E}\Psi' = \delta(\text{E}\Psi, \text{E}Z)$ and $\Psi \neq \emptyset$, and thus $\dots \subseteq \Psi_{k+2} \subseteq \Psi_{k+1} \subseteq \Psi_k$. Since $\Psi_k \subseteq \text{Cl}(\varphi_1)$ is finite, there must be $j \geq k$ and $\Psi \neq \emptyset$ such that $\Psi_i = \Psi$ for all $i \geq j$. Pick any $\psi \in \Psi$, then we find that there are $\varphi, \varphi' \in \{\psi, \text{X}\psi\}$ such that $(\varphi, \varphi') \in Z_j$. By Proposition 4.13 there is a trace $\varphi_0, \dots, \varphi_{j-1}$ through Z_1, \dots, Z_{j-1} with $\varphi_{j-1} = \varphi$. Define $\varphi_j = \varphi'$. Once we have defined $\varphi_0, \dots, \varphi_i$ and have $\psi \in \Psi_i$, we get that $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ for some $\varphi_{i+1} \in \{\psi, \text{X}\psi\}$, and therefore $\psi \in \Psi_{i+1}$. We continue this way and construct a trace $\varphi_0, \varphi_1, \varphi_2, \dots$ through Z_1, Z_2, \dots which ends up in a U-cell. Therefore Acc rejects the sequence.

Suppose $Q = \text{A}$ and Acc accepts the sequence. This means that there is a trace $\varphi_0, \varphi_1, \varphi_2, \dots$ that ends up in a R-cell; let $k \geq 1$ and ψ be such that ψ is an R-formula

and $\varphi_i \in \{\psi, X\psi\}$ for all $i \geq k$. Let $Q_1\emptyset, A\Psi_1, A\Psi_2, \dots$ be the run of \mathbb{B}_ξ on the sequence. If $\Psi_i \neq \emptyset$ for all $i \geq k$, then \mathbb{B}_ξ accepts the sequence. So assume that there is $j \geq k$ such that $\Psi_j = \emptyset$. If $\Psi_i = \emptyset$ then either the rule for $A\Psi_i$ is not modal and $\Psi_{i+1} = \emptyset$, or $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ and $(\varphi_i, \varphi_{i+1}) = (X\psi, \psi)$, hence $\psi \in \Psi_{i+1}$. Since the trace is non-stalling, there is $j' > j$ with $\psi \in \Psi_{j'}$. If $i \geq j'$ and $\psi \in \Psi_i$, then from $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ we get that $\psi \in \Psi_{i+1}$. We conclude that $\psi \in \Psi_i$ for all $i \geq j'$, and therefore that $\Omega(A\Psi_i) = 0$ for all $i \geq j'$. Thus \mathbb{B}_ξ accepts the sequence.

Lastly, suppose $Q = A$ and \mathbb{B}_ξ accepts the sequence. Let $Q_1\emptyset, A\Psi_1, A\Psi_2, \dots$ be the run of \mathbb{B}_ξ on the sequence, then there is $k \geq 1$ such that $\Psi_i \neq \emptyset$ for all $i \geq k$. Observe that $\Psi' \subseteq \Psi$ if $A\Psi' = \delta(A\Psi, AZ)$ and $\Psi \neq \emptyset$, and thus $\dots \subseteq \Psi_{k+2} \subseteq \Psi_{k+1} \subseteq \Psi_k$. Since $\Psi_k \subseteq \text{Cl}(\varphi_1)$ is finite, there must be $j \geq k$ and $\Psi \neq \emptyset$ such that $\Psi_i = \Psi$ for all $i \geq j$. Pick any $\psi \in \Psi$, then we find that there are $\varphi, \varphi' \in \{\psi, X\psi\}$ such that $(\varphi, \varphi') \in Z_j$. By Proposition 4.13 there is a trace $\varphi_0, \dots, \varphi_{j-1}$ through Z_1, \dots, Z_{j-1} with $\varphi_{j-1} = \varphi$. Define $\varphi_j = \varphi'$. Once we have defined $\varphi_0, \dots, \varphi_i$ and have $\psi \in \Psi_i$, we get that $(\varphi_i, \varphi_{i+1}) \in Z_{i+1}$ for some $\varphi_{i+1} \in \{\psi, X\psi\}$, and therefore $\psi \in \Psi_{i+1}$. We continue this way and construct a trace $\varphi_0, \varphi_1, \varphi_2, \dots$ through Z_1, Z_2, \dots which ends up in an R-cell. Therefore Acc accepts the sequence. \square

With \mathbb{B}_ξ in place, we can create an automaton $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ that simultaneously runs both \mathbb{A}_ξ and \mathbb{B}_ξ . Its states are quantified continuations from A_ξ that are annotated with a to-do list from B_ξ . The transition term for (a, b) is based on the transition term of a as defined by \mathbb{A}_ξ , and the priority of (a, b) will be given by that of b .

Definition 4.15. Let ξ be a CTL_Σ^* formula, let $\mathbb{A}_\xi = (A_\xi, \Delta, \text{Acc}, a_1)$ be its modal automaton and let $\mathbb{B}_\xi = (B_\xi, \delta, \Omega, b_1)$ be its A_ξ -stream parity automaton. The modal parity automaton $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ is given by $(A_\xi \times B_\xi, \Delta^\odot, \Omega^\odot, (a_1, b_1))$ where $\Delta^\odot(a, b)$ is created from $\Delta(a)$ by replacing occurrences of states a' with $(a', \delta(b, a))$, and where $\Omega^\odot(a, b) := \Omega(b)$. \triangleleft

The proof of its correctness is fairly standard.

Proposition 4.16. Let ξ be a CTL_Σ^* formula, let \mathbb{A}_ξ be its modal automaton and let \mathbb{B}_ξ be its A_ξ -stream parity automaton. Let $(a, b) \in A_\xi \times B_\xi$ then $(\mathbb{A}_\xi, a) \equiv (\mathbb{A}_\xi \odot \mathbb{B}_\xi, (a, b))$. In particular $\mathbb{A}_\xi \equiv \mathbb{A}_\xi \odot \mathbb{B}_\xi$.

Proof. Note that if $(a, b) \rightarrow (a', b')$ and $(a, b) \rightarrow (a'', b'')$, then $b' = \delta(b, a) = b''$. This means that every sequence $(a_0, b_0), (a_1, b_1), \dots$ with $(a_i, b_i) \rightarrow (a_{i+1}, b_{i+1})$ in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ is uniquely determined by b_0 and the sequence a_0, a_1, \dots with $a_i \rightarrow a_{i+1}$ in \mathbb{A}_ξ . Given $(a_0, b_0) \in A_\xi \times B_\xi$ we therefore have a one-to-one correspondence between sequences in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ that start from (a_0, b_0) and sequences in \mathbb{A}_ξ that start from a_0 . Since the transition terms $\Delta^\odot(a_i, b_i)$ and $\Delta(a_i)$ are the same with respect this correspondence, we can translate matches of the acceptance game of $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ to that of \mathbb{A}_ξ , and vice versa. It remains to check that the winning conditions for infinite matches are the same, but this is given by Proposition 4.14. \square

Because we have not changed the transition terms in any meaningful way, dominated clusters of \mathbb{A}_ξ correspond to dominated clusters in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$.

Proposition 4.17. *For any $\xi \in \text{CTL}_{\Sigma}^*$ the automaton $\mathbb{A}_{\xi} \odot \mathbb{B}_{\xi}$ has dominated clusters.*

Proof. Let $(a, b) \in A_{\xi} \times B_{\xi}$. Let C be the cluster of \mathbb{A}_{ξ} that a belongs to. From the construction it follows that the cluster of $\mathbb{A}_{\xi} \odot \mathbb{B}_{\xi}$ that (a, b) belongs to is a subset of the generalized cluster $C \times B_{\xi}$. By Proposition 4.8, C is dominated. Now if C is \exists -dominated, then $C \times B_{\xi}$ is \exists -dominated as well. Therefore the cluster that (a, b) belongs to is also \exists -dominated. Similarly if C is \forall -dominated, then so is the cluster that (a, b) belongs to. \square

We are left to make the automaton $\mathbb{A}_{\xi} \odot \mathbb{B}_{\xi}$ guarded. Turning arbitrary (silent-step) modal automata into guarded modal automata is a bit of a challenge and usually results in an exponential blowup in the size of the automata. Here we will give two properties that allow this guardification to be relatively painless; in fact, they shrink the automaton instead. First, let us separate guarded and unguarded transitions.

Definition 4.18. Let \mathbb{A} be a modal automaton. Define \rightarrow_{\circ} and \rightarrow_{τ} on A by $a \rightarrow_{\circ} b$ (resp. $a \rightarrow_{\tau} b$) whenever b occurs guarded (resp. unguarded) in $\Delta(a)$. \triangleleft

The basic idea of guardification is that we replace unguarded occurrences of a state a with its transition term $\Delta(a)$. However this transition term might in turn contain unguarded occurrences of other states. To avoid getting stuck in an infinite loop, we demand that the automata do not contain silent-cycles.

Definition 4.19. A modal automaton \mathbb{A} is *silent-cycle safe* if \rightarrow_{τ} is acyclic, i.e. if there is no sequence $a_0 \rightarrow_{\tau} \dots \rightarrow_{\tau} a_n \rightarrow_{\tau} a_0$ in \mathbb{A} . \triangleleft

We have already seen that the automata constructed in section 4.1 have this property.

Proposition 4.20. *The automaton \mathbb{A}_{ξ} is silent-cycle safe for any ξ .*

Proof. Let ic be the immediate complexity on \mathbb{A}_{ξ} defined in Definition 4.9. Analyzing the Dam rules gives us that if $a \rightarrow b$ in \mathbb{A}_{ξ} , then $\text{ic}(a) > \text{ic}(b)$ unless the rule applied to a is a modal rule. Let $a_0 \rightarrow_{\tau} \dots \rightarrow_{\tau} a_n \rightarrow_{\tau} a_{n+1}$ in \mathbb{A}_{ξ} then $\text{ic}(a_0) > \dots > \text{ic}(a_n) > \text{ic}(a_{n+1})$, thus it cannot be that $a_{n+1} = a_0$. Therefore \mathbb{A}_{ξ} is silent-cycle safe. \square

For automata that have this property, defining a guarded modal automaton that looks similar is easy.

Definition 4.21. Let $\mathbb{A} = (A, \Delta, \Omega, a_{\perp})$ be a modal parity automaton that is silent-cycle safe. Since A is finite, \rightarrow_{τ} is converse-wellfounded. Thus we can define the silent-step depth $\text{ssd} : A \rightarrow \mathbb{N}$ such that $\text{ssd}(a)$ is the length of the longest sequence $a_1, \dots, a_n \in A$ such that $a_i \rightarrow_{\tau} a_{i+1}$ for all $i < n$. Clearly $\text{ssd}(a) \geq \text{ssd}(b) + 1$ whenever $a \rightarrow_{\tau} b$, i.e. whenever b occurs unguarded in $\Delta(a)$. The map $\Delta^{\gamma} : A \rightarrow \text{gML}_1(A)$ is defined recursively using ssd . Let $a \in A$ with $\text{ssd}(a) = k$ and assume that $\Delta^{\gamma}(b)$ has been defined for all $b \in A$ with $\text{ssd}(b) < k$. Create $\Delta^{\gamma}(a)$ from $\Delta(a)$ by replacing every unguarded occurrence of a state b with its guarded transition term $\Delta^{\gamma}(b)$. Now define $\mathbb{A}^{\gamma} := (A, \Delta^{\gamma}, \Omega, a_{\perp})$. \triangleleft

In fact for finite matches, this automaton is equivalent. For infinite matches however, the trace of the original automaton might contain states with a high priority that are “skipped” in the guarded automaton. Again, we avoid this problem by demanding that it is safe to skip the priorities of states that are reached by silent steps.

Definition 4.22. A modal parity automaton \mathbb{A} is *silent-priority safe* if for every sequence $a_0 \rightarrow_\tau \dots \rightarrow_\tau a_n \rightarrow_\circ a_{n+1}$ in \mathbb{A} where a_0 and a_{n+1} belong to the same cluster, it holds that $\Omega(a_i) \leq \Omega(a_{n+1})$ for all $i > 0$. \triangleleft

Note that these two properties are very strong; most of the work of guardifying a modal automaton usually lies in obtaining such properties and that is where the exponential blowup occurs. However we constructed the automaton $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ with this property in mind; the priority of a state in $A_\xi \times B_\xi$ only decreases when a new to-do list is created, and this is only done directly after a modal step.

Proposition 4.23. *For any CTL $^*_\Sigma$ formula ξ the modal parity automaton $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ is both silent-cycle safe and silent-priority safe.*

Proof. The silent-cycle safeness follows from that of \mathbb{A}_ξ , which is proven in Proposition 4.20.

For the silent-priority safeness, let $(a_0, b_0) \rightarrow_\tau \dots \rightarrow_\tau (a_n, b_n) \rightarrow_\circ (a_{n+1}, b_{n+1})$ in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ such that (a_0, b_0) and (a_{n+1}, b_{n+1}) belong to the same cluster. Since $(a_0, b_0) \twoheadrightarrow (a_i, b_i)$ and $(a_i, b_i) \twoheadrightarrow (a_{n+1}, b_{n+1})$, all the (a_i, b_i) belong to that cluster. From Proposition 4.7 it follows that there is Q such that $a_i = QZ_i$ and $b_i = Q\Psi_i$ for each i . If $Q = \mathbf{E}$ then $\Omega(a_i, b_i) \in \{1, 2\}$ for all i ; if $Q = \mathbf{A}$ then $\Omega(a_i, b_i) \in \{0, 1\}$ for all i . Let $u := 1$ in the former case and $u := 0$ in the latter, then $\Omega(a_i, b_i) = u + 1$ if $\Psi_i = \emptyset$ and $\Omega(a_i, b_i) = u$ otherwise. If $\Omega(a_i, b_i) = u$ for all $i > 0$ then $\Omega(a_{n+1}, b_{n+1}) \geq u$ and we are done. So suppose there is $j \in \{1, \dots, n\}$ with $\Psi_j = \emptyset$. Since $(a_{j-1}, b_{j-1}) \rightarrow_\tau (a_j, b_j)$, the rule for QZ_{j-1} was not a modal rule, and therefore Z_j contains no pairs of the form $(\mathbf{X}\varphi, \varphi)$. This means $\Psi_{j+1} = \emptyset$. Therefore if $\Psi_j = \emptyset$ with $j > 0$ then $\Psi_i = \emptyset$ for all $i \geq j$; in particular $\Psi_{n+1} = \emptyset$ and hence $\Omega(a_{n+1}, b_{n+1}) = u + 1$. \square

Thus it remains to prove that guardification works for automata that are both silent-cycle safe and silent-priority safe.

Proposition 4.24. *For every modal parity automaton \mathbb{A} that is both silent-cycle safe and silent-priority safe, the guarded modal parity automaton \mathbb{A}^γ is equivalent to \mathbb{A} , has at most as many states as \mathbb{A} and has a Mostowski index inside that of \mathbb{A} .*

Proof. Let $\mathbb{A} = (A, \Delta, \Omega, a_1)$ be such an automaton. We will show that $(\mathbb{A}, a) \equiv (\mathbb{A}^\gamma, a)$ for all $a \in A$, so that in particular $\mathbb{A} \equiv \mathbb{A}^\gamma$.

Let $a \in A$. Let \mathbb{S} be a transition system and let $s \in S$. Suppose $\mathbb{S}, s \Vdash (\mathbb{A}, a)$, then \exists has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}^\gamma, \mathbb{S})@(s, a)$, \exists plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$ that is consistent with her strategy. Whenever the shadow match is at a position (s, φ) the real match will be at the position (s, φ^γ) , where $\varphi^\gamma \in \text{gML}_1(A)$ is obtained from φ by replacing every unguarded occurrence of a state a

with $\Delta^\gamma(a)$. If the shadow match moves from (s, φ) to (s, ψ) where ψ is a subformula of φ , then in the real match \exists can move from (s, φ^γ) to (s, ψ^γ) . If the shadow match moves from (s, a) to $(s, \Delta(a))$, then the real match will already be at $(s, (\Delta(a))^\gamma) = (s, \Delta^\gamma(a))$.

In this way the two matches will be nearly identical, with the exception that the real match might skip some unguarded positions (s, a) ; that is, if the shadow match contains a sequence $\dots, (s, \varphi), (s, a), (s, \Delta(a)), \dots$ where a occurs unguarded in φ , then the real match will have the sequence $\dots, (s, \varphi^\gamma), (s, \Delta^\gamma(a)), \dots$ instead. Of course finite matches are won by the same player. So suppose that both matches are infinite, then because the shadow match is consistent with f , the highest priority occurring infinitely often in the shadow match, say m , is even. Let a_0, a_1, \dots be the sequence of states visited in the shadow match, then we may assume that they all belong to the same cluster. The states visited in the real match will be a_{e_0}, a_{e_1}, \dots where each a_{e_i} occurred guarded in the transition term of $a_{e_{i-1}}$. For any omitted unguarded state a_i with $\Omega(a_i) = m$ we can find i' and i'' with $i' < i \leq i''$ such that $a_{i'} \rightarrow_\tau \dots \rightarrow_\tau a_{i''} \rightarrow_\circ a_{i''+1}$. Now the silent-priority safeness of \mathbb{A} guarantees that $\Omega(a_i) \leq \Omega(a_{i''+1})$ hence $\Omega(a_{i''+1}) = m$. This means that for every i with $\Omega(a_i) = m$ there is j such that $e_j \geq i$ and $\Omega(a_{e_j}) = m$. Therefore the highest priority occurring infinitely often in the real match is also m , thus \exists wins and $\mathbb{S}, s \Vdash (\mathbb{A}^\gamma, a)$.

For the converse, suppose $\mathbb{S}, s \not\Vdash (\mathbb{A}, a)$. By Proposition 2.43, this means \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$ and we need to show that \forall has a winning strategy for $\mathcal{A}(\mathbb{A}^\gamma, \mathbb{S})@(s, a)$. The rest of the proof is dual. \square

Since we want to work with automata with dominated clusters, it is important that this property is preserved under this operation. For guardification this is indeed the case.

Proposition 4.25. *If \mathbb{A} has dominated clusters then so does \mathbb{A}^γ .*

Proof. In the construction of \mathbb{A}^γ , the modalities that occur are not changed in any way. Each \exists -dominated cluster in \mathbb{A} corresponds to a \exists -dominated cluster in the modified automaton, and the same holds for \forall -dominated clusters. Note that for $a, b \in A$ with $a \rightarrow b$, if c occurs in $\Delta(b)$ then $c \not\rightarrow a$. \square

We conclude that for every CTL_Σ^* formula ξ , the automaton $(\mathbb{A}_\xi \odot \mathbb{B}_\xi)^\gamma$ is a guarded modal parity automaton with dominated clusters. Note that in the construction of \mathbb{B}_ξ the highest priority used is 2. In fact, \exists -dominated clusters use the priorities 1 and 2 whereas \forall -dominated clusters use the priorities 0 and 1. The following theorem summarizes the results of this chapter.

Theorem 4.26. *For every CTL_Σ^* formula there is an equivalent guarded modal parity automaton that has dominated clusters, whose \exists -dominated clusters are Büchi and whose \forall -dominated clusters are co-Büchi.*

Proof. Let ξ be a CTL_Σ^* formula. By Lemma 4.11 we have $\xi \equiv \mathbb{A}_\xi$, and by Proposition 4.16 we have $\mathbb{A}_\xi \equiv \mathbb{A}_\xi \odot \mathbb{B}_\xi$. By Proposition 4.23 we can use Proposition 4.24 to get an equivalent guarded modal parity automaton $(\mathbb{A}_\xi \odot \mathbb{B}_\xi)^\gamma$. Lastly we use Proposition 4.17 and Proposition 4.25 to get that this automaton has dominated clusters. \square

Chapter 5

Dominance in the modal μ -calculus and its automata

In chapter 4, we give a translation of CTL* into guarded modal parity automata with dominated clusters. Unfortunately this class of automata is too broad to allow a translation back into CTL*. As a minimal counter-example, consider the automaton $\mathbb{A} = (A, \Delta, \Omega, a)$ where $A = \{a, b\}$, $\Delta(a) = p \wedge \diamond b$, $\Delta(b) = \diamond a$, $\Omega(a) = 2$ and $\Omega(b) = 2$. Now \mathbb{A} is a guarded modal parity automaton and its only cluster is both \exists -dominated and Büchi. We will discuss why this automaton cannot be expressed by a CTL* formula in chapter 7. In the modal μ -calculus, however, this automaton can be expressed by the formula $\nu x.(p \wedge \diamond \diamond x)$.

In this chapter, we will explore what dominance means in the context of fixpoints, and introduce a fragment of μ ML that characterizes the class of modal automata with dominated clusters.

5.1 Automata and the modal μ -calculus

In this section, we will construct modal automata for μ ML formulas and μ ML formulas for modal automata. Although standard translations already exist due to Janin and Walukiewicz (1995) and others, we will give slightly modified constructions here in order to better fit the needs of section 5.2, where we will show that the translations preserve dominance. In particular the translation from automata to clean formulas is very meticulous, which will pay off later in the chapter.

When translating between μ ML and modal automata, it is convenient to first find a guarded equivalent. To understand what guardedness means for a μ ML formula, we first generalize the modal one-step formulas from Definition 2.36 to basic modal logic formulas over a set A .

Definition 5.1. The syntax of $\text{ML}(A)$ is generated by the grammar

$$\varphi ::= \ell \mid a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \diamond \varphi \mid \square \varphi$$

where $\ell \in \text{Lit}$ and $a \in A$. An occurrence of $a \in A$ in a modal formula φ is \diamond -*guarded* if it is part of some subformula $\diamond\psi$ of φ . Similarly it is \square -*guarded* if it is within the scope of a \square . It is *unguarded* if it is neither \diamond -guarded nor \square -guarded. \triangleleft

Now a μML formula is guarded if for every subformula $\lambda x.\vartheta$, x does not occur unguarded in ϑ . It can be shown that every μML formula is equivalent to a guarded μML formula. In this paper, however, we will use a particular fragment of μML which we will call the modal “one-step” μ -calculus. The formulas of $\mu_1\text{ML}$ already look a lot like the modal one-step formulas found in modal automata. We will give translations between $\mu_1\text{ML}$ and modal automata. Together with the standard translations between μML and modal automata, this shows that restricting to $\mu_1\text{ML}$ does not decrease the expressive power of μML .

Definition 5.2. The syntax of $\mu_1\text{ML}(Y)$ is generated by the dual grammar

$$\begin{aligned}\varphi &::= \ell \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \diamond\psi \mid \square\psi \\ \psi &::= y \mid \mu x.\vartheta \mid \nu x.\vartheta\end{aligned}$$

where $\ell \in \text{Lit}$, $y \in Y$, $x \in \text{Var}$ and $\vartheta \in \mu_1\text{ML}_M(Y \cup \{x\})$. The set $\mu_1\text{ML}_M(Y)$ of modal formulas of $\mu_1\text{ML}(Y)$ is generated by φ . The set $\mu_1\text{ML}_\Lambda(Y)$ of point-formulas of $\mu_1\text{ML}(Y)$ is generated by ψ . The syntax of $\mu_1\text{ML}$ is given by $\mu_1\text{ML}_\Lambda(\emptyset)$. \triangleleft

Note that $\mu_1\text{ML}$ formulas are always guarded. The idea behind the phrase “one-step” in “modal one-step μ -calculus” is that every $\mu_1\text{ML}_M$ formula looks like a guarded modal one-step formula when we view $\mu_1\text{ML}_\Lambda$ formulas as objects. In general we can decompose a μML_M formula – i.e. a formula that isn’t a free variable or fixpoint formula – as a modal formula over a set of μML_Λ formulas – i.e. over a set of free variables and fixpoint formulas.

Definition 5.3. Every $\mu\text{ML}_M(Y)$ formula φ has a canonical *decomposition* of the form $\varphi = \alpha(\psi_1, \dots, \psi_n)$ where $\alpha \in \text{ML}(\mathbb{N})$ and where each $\psi_i \in \mu\text{ML}_\Lambda(Y)$. \triangleleft

The natural numbers here act as temporary meta-variables. As a concrete example, the formula $(p \wedge \square x) \vee \mu y.(\square y \vee q)$ is decomposed as $\alpha(x, \mu y.(\square y \vee q))$ where $\alpha = (p \vee \square 1) \vee 2$. Now when a $\mu_1\text{ML}_M$ formula φ is decomposed as $\alpha(\psi_1, \dots, \psi_n)$, α will be a guarded modal one-step formula.

Decomposition gives a natural way to define automata for $\mu_1\text{ML}_\Lambda$ formulas. There is a one-to-one correspondence between fixpoint formulas in $\text{Sb}(\xi)$ and $\text{BV}(\xi)$; for every $x \in \text{BV}(\xi)$ we have $\lambda_x x.\vartheta_x \in \text{Sb}(\xi)$ and for every $\lambda x.\vartheta \in \text{Sb}(\xi)$ we have $\lambda = \lambda_x$ and $\vartheta = \vartheta_x$. Therefore we identify bound variables with their binding formulas, so that when either x or $\lambda_\xi(x)$ occurs in ϑ_y there is a transition from y to x ; or rather, from the state \underline{y} to the state \underline{x} . This will be the basis for the transition terms.

We still need a winning condition for infinite matches. Here we use the similarity between the winning conditions of the evaluation game for μML , Definition 2.26, and the acceptance game for modal automata, Definition 2.40. For infinite matches in both games there is a greatest object (variable, state) that is visited infinitely often, and this object is

either “good” (bound by ν , even parity) or “bad” (bound by ν , odd parity) for \exists . Thus, the parity for the state \underline{x} will be even when x is bound by a ν -formula, and odd when it is bound by a ν -formula. This still leaves the ordering of the priorities, but this problem was already solved in the proof of Proposition 2.29.

Definition 5.4. Let ξ be a $\mu_1\text{ML}_\Lambda$ formula. Define $A_\xi := \{\underline{x} \mid x \in \text{BV}(\xi)\}$. Let $x \in \text{BV}(\xi)$, then we can decompose ϑ_x as $\vartheta_x = \alpha(y_1, \dots, y_n, \lambda_\xi(z_1), \dots, \lambda_\xi(z_k))$ where $\alpha \in \text{gML}_1(\mathbb{N})$ and $y_1, \dots, y_n, z_1, \dots, z_k \in \text{BV}(\xi)$. Define $\Delta_\xi(\underline{x}) := \alpha(\underline{y}_1, \dots, \underline{y}_n, \underline{z}_1, \dots, \underline{z}_k)$ and $\Omega_\xi(\underline{x}) = \Omega_\xi(x)$. Define $\mathbb{A}_\xi := (A_\xi, \Delta_\xi, \Omega_\xi, \underline{x})$ where x is the variable bound by ξ . \square

In the following, we will be a bit more lax with notation and omit ξ from $\lambda_\xi(x)$. The clusters of \mathbb{A}_ξ correspond to the \bowtie_ξ equivalence classes of $\text{Sb}(\xi)$ in a natural way when identifying states \underline{x} with their binding formulas $\lambda(x)$.

Proposition 5.5. Let $\xi \in \mu_1\text{ML}_\Lambda$. For all $x, y \in \text{BV}(\xi)$, $\lambda(x) \triangleright_\xi \lambda(y)$ iff $\underline{x} \rightarrow \underline{y}$ in \mathbb{A}_ξ .

Proof. Let $x, y \in \text{BV}(\xi)$ with $\lambda(x) \triangleright_\xi \lambda(y)$, then there are $\varphi_0, \dots, \varphi_n$ such that $\varphi_0 = \lambda(x)$, $\varphi_n = \lambda(y)$ and $\varphi_{i+1} \in \nabla_\xi(\varphi_i)$. Let i_0, \dots, i_k be the indices of fixpoint formulas, then there are $z_0, \dots, z_k \in \text{BV}(\xi)$ such that $z_0 = x$, $z_k = y$ and $\varphi_{i_j} = \lambda(z_j)$ for each j . Let $j < k$, then there are no fixpoint formulas between φ_{i_j} and $\varphi_{i_{j+1}}$, and the only possible bound variable would be z_{j+1} . As such, either z_{j+1} or $\lambda(z_{j+1})$ occurs in the decomposition of $\varphi_{i_{j+1}}$, hence \underline{z}_{j+1} occurs in $\Delta_\xi(\underline{z}_j)$. This means $\underline{z}_j \rightarrow \underline{z}_{j+1}$ for all j , which gives $\underline{x} \rightarrow \underline{y}$.

Let $x, y \in \text{BV}(\xi)$ with $\underline{x} \rightarrow \underline{y}$, then there are z_0, \dots, z_n such that $z_0 = x$, $z_n = y$ and $\underline{z}_i \rightarrow \underline{z}_{i+1}$. From the construction of Δ_ξ we see that this means that for each i either z_{i+1} or $\lambda(z_{i+1})$ is a subformula of $\lambda(z_i)$. Since $u \triangleright_\xi \lambda(u)$ for all $u \in \text{BV}(\xi)$ we get $\lambda(z_i) \triangleright_\xi \lambda(z_{i+1})$ for all i . This gives $\lambda(x) \triangleright_\xi \lambda(y)$. \square

Proposition 5.6. Let $\xi \in \mu_1\text{ML}_\Lambda$. For all $x, y \in \text{BV}(\xi)$, $\lambda(x) \bowtie_\xi \lambda(y)$ iff \underline{x} and \underline{y} belong to the same cluster of \mathbb{A}_ξ .

Proof. Let $x, y \in \text{BV}(\xi)$. By Proposition 5.5 we have that $\lambda(x) \bowtie_\xi \lambda(y)$ iff $\lambda(x) \triangleright_\xi \lambda(y)$ and $\lambda(y) \triangleright_\xi \lambda(x)$ iff $\underline{x} \rightarrow \underline{y}$ and $\underline{y} \rightarrow \underline{x}$ iff \underline{x} and \underline{y} belong to the same cluster. \square

The following theorem states that this automaton construction is indeed correct.

Theorem 5.7. For every $\mu_1\text{ML}_\Lambda$ formula ξ there is an equivalent guarded modal parity automaton \mathbb{A}_ξ .

Proof. Let $\xi \in \mu_1\text{ML}_\Lambda$ and let x_0 be such that $\lambda(x_0) = \xi$. Let \mathbb{S} be a transition system and let $s_I \in S$. The evaluation game $\mathcal{E}(\xi, \mathbb{S})@(s_I, \xi)$ and the acceptance game $\mathcal{A}(\mathbb{A}_\xi, \mathbb{S})@(s_I, \underline{x}_0)$ are very similar. Suppose a match of each game is played, one a real match and the other a shadow match. The evaluation match starts at $(s, \lambda(x_0))$ and the acceptance match starts at (s, \underline{x}_0) .

If the evaluation match is at $(s, \lambda_x x. \vartheta_x)$ and the acceptance match is at (s, \underline{x}) , the evaluation match continues to $(s, \vartheta_x) = (s, \alpha(y_1, \dots, y_n, \lambda(z_1), \dots, \lambda(z_k)))$ for some $\alpha \in \text{gML}_1(\mathbb{N})$ and some $y_1, \dots, y_n, z_1, \dots, z_k \in \text{BV}(\xi)$. Meanwhile the acceptance match

continues to $(s, \Delta_\xi(\underline{x})) = (s, \alpha(\underline{y}_1, \dots, \underline{y}_n, \underline{z}_1, \dots, \underline{z}_n))$. Again, moves in the former can be mimicked with moves in the latter and vice versa, until a literal is reached in both games, in which case the winner is the same, or until a pairing (t, y_i) and (t, \underline{y}_i) for some $i \leq n$ is reached, or until a pairing $(t, \lambda(z_i))$ and (t, \underline{z}_i) for some $i \leq k$ is reached.

If the evaluation match is at (s, x) and the acceptance match is at (s, \underline{x}) , then the evaluation match continues to $(s, \lambda_x x. \vartheta_x)$ and we arrive at the previous pairing.

It remains to check that the winner is the same in the case that the matches are infinite; in this case the evaluation match has a \preceq -greatest bound variable x that it goes through infinitely often, and based on the pairings the acceptance match will go through the state \underline{x} infinitely often. Let $\Omega_\xi(\underline{y})$ be the highest priority among states visited infinitely often, then $\Omega_\xi(\underline{y}) = \Omega_\xi(y) \leq \Omega_\xi(x)$ because $\lambda(y)$ is a subformulas of $\lambda(x)$, and thus $\Omega_\xi(\underline{y}) = \Omega_\xi(\underline{x})$. Since $\Omega_\xi(\underline{x}) = \Omega_\xi(x)$ is even if and only if $\lambda_x = \nu$, the winner of both games is the same. \square

Now we also want a converse to Theorem 5.7, that is, we want to translate modal automata into clean $\mu_1\text{ML}_\Lambda$ formulas. As a simple example, consider the automaton (A, Δ, Ω, a) where $A = \{a, b\}$, $\Delta(a) = \diamond b \vee (p \wedge \diamond a)$, $\Delta(b) = \Box a \vee (q \wedge \diamond b)$, $\Omega(a) = 2$ and $\Omega(b) = 1$. With the translation from formulas to automata in mind, states with even priorities should turn into ν -formulas and states with odd priorities into μ -formulas. Thus we might construct for a the formula $\varphi_a = \nu x_a. (\diamond x_b \vee (p \wedge \diamond x_a))$ and for b the formula $\varphi_b = \mu x_b. (\Box x_a \vee (q \wedge \diamond x_b))$. Indeed the formula $\psi = \nu x_a. (\diamond \mu x_b. (\Box x_a \vee (q \wedge \diamond x_b)) \vee (p \wedge \diamond x_a))$, where we replaced x_b in φ_a with φ_b , is equivalent to (A, Δ, Ω, a) . The nesting of the variables here works because $x_b \prec x_a$ and $\Omega(b) < \Omega(a)$. If we applied the same method to (A, Δ, Ω, b) however, we would replace x_a in φ_b with φ_a and get $\mu x_b. (\Box (\nu x_a. (\diamond x_b \vee (p \wedge \diamond x_a))) \vee (q \wedge \diamond x_b))$. This is not equivalent, because here $x_a \prec x_b$. Instead we could take

$$\mu x'_b. (\Box (\nu x_a. (\diamond \mu x_b. (\Box x_a \vee (q \wedge \diamond x_b)) \vee (p \wedge \diamond x_a))) \vee (q \wedge \diamond x'_b))$$

where we first replace every x_b in φ_b with a fresh variable x'_b and then every x_a with ψ . Now $x_b \prec x_a \prec x'_b$; the variable x'_b does not occur in ψ hence $x_a \prec x'_b$ is not a problem.

This already shows that the converse direction is a bit more involved. States can be represented by multiple fixpoint formulas, depending on which other states are allowed to appear as variables and which states must be represented by new fixpoint formulas. States with lower priority are always represented by fixpoint formulas; it is the states with higher priority that can cause problems. As such, the variables used are indexed by a state together with a set of states that have higher priority. Put differently, the variables are indexed by a set of states and represent the state with the least priority. Because priorities only matter inside a cluster, these sets of states are always subsets of a cluster.

Definition 5.8. Let $\mathbb{A} = (A, \Delta, \Omega, a_I)$ be a guarded modal parity automaton. By induction on clusters we define a formula $\xi_a \in \mu_1\text{ML}$ for every $a \in A$.

Let C be a cluster of \mathbb{A} . The induction hypothesis states that whenever $a \in C$ and $a \rightarrow b$ either $b \in C$ or ξ_b has been defined. Fix a numbering $C = \{c_1, \dots, c_n\}$ such that $\Omega(c_i) \leq \Omega(c_j)$ whenever $i \leq j$. Define $\mathcal{I} := \{I \subseteq \{1, \dots, n\} \mid I \neq \emptyset\}$. For $I \in \mathcal{I}$ we write $c_I := c_{\min(I)}$. For $I \in \mathcal{I}$ and $j \in \{1, \dots, n\}$, we write $I(j) := \{i \in I \mid i \geq j\}$. We

assume we have a set of fresh variables $X := \{x_I \mid I \in \mathcal{I}\} \subseteq \text{Var}$. Let r be the bijective map $r : \mathcal{I} \rightarrow \{1, \dots, 2^n - 1\}$ by $r(I) := 2^n - \sum_{i \in I} 2^{i-1}$, then $r(I)$ is the rank of I . Note that $r(I) \geq r(J)$ whenever $I \subseteq J$ and that $r(I(j) \cup \{j\}) < r(I)$ whenever $j \notin I$. Using induction on r , we will define a fixpoint formula $\varphi_I \in \mu_1\text{ML}(X)$ for every $I \in \mathcal{I}$.

For $I \in \mathcal{I}$, define $\varphi_I := \lambda_I x_I \vartheta_I$, where $\lambda_I := \nu$ if $\Omega(c_I)$ is even and $\lambda_I := \mu$ if $\Omega(c_I)$ is odd, and where ϑ_I is created from $\Delta(c_I)$ by replacing every $b \in A \setminus C$ with ξ_b , every c_i where $i \in I$ with $x_{I(i)}$ and every c_j where $j \notin I$ with $\varphi_{I(j) \cup \{j\}}$. Note that ϑ_I is a $\mu_1\text{ML}$ formula because $\Delta(c_I)$ is guarded. Also note that $\text{FV}(\vartheta_I) \subseteq \{x_{I(i)} \mid i \in I\}$. Lastly note that φ_I is clean and that if φ_I is a proper subformula of φ_J , then $r(I) < r(J)$.

Let $k \in \{1, \dots, n\}$, then define $\xi_{c_k} := \varphi_{\{k\}} \in \mu_1\text{ML}(\emptyset)$. \triangleleft

Note that the set \mathcal{I} , the variables from X , the ranking r etcetera are all relative to the cluster C . To avoid even more laborious notation, we will leave this implicit. This poses no problems for the proof of Theorem 5.9 because it is based on induction on cluster-depth.

Theorem 5.9. *For every guarded modal parity automaton \mathbb{A} there is an equivalent $\mu_1\text{ML}_{\Lambda}$ formula $\xi_{\mathbb{A}}$.*

Proof. Let $\mathbb{A} = (A, \Delta, \Omega, a_I)$ be a guarded modal parity automaton. By induction on cluster-depth we show that $(\mathbb{A}, a) \equiv \xi_a$ for all $a \in A$. This then gives $\mathbb{A} \equiv \xi_{\mathbb{A}} := \xi_{a_I}$.

Let C be a cluster of \mathbb{A} . The induction hypothesis states that whenever $a \in C$ and $a \rightarrow b$ either $b \in C$ or $\xi_b \equiv b$ has already been shown. Let $k \in \{1, \dots, n\}$. We need to show that $\xi_{c_k} \equiv (\mathbb{A}, c_k)$. Let \mathbb{S} be a transition system and let $s_I \in S$. Suppose that $\mathbb{S}, s_I \Vdash \varphi_{\{k\}}$, then \exists has a winning strategy f for $\mathcal{E}(\varphi_{\{k\}}, \mathbb{S})@_{(s_I, \varphi_{\{k\}})}$. By Proposition 2.29 we may assume that f is positional. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@_{(s_I, c_k)}$, \exists plays a shadow match of $\mathcal{E}(\varphi_{\{k\}}, \mathbb{S})@_{(s_I, \varphi_{\{k\}})}$ consistent with f , in such a way that positions (s, c_I) in the real match correspond to positions (s, φ_I) in the shadow match.

Suppose the real match is at (s, c_I) and the shadow match is at (s, φ_I) , then the matches necessarily continue to $(s, \Delta(c_I))$ and (s, ϑ_I) respectively. By the construction of ϑ_I , moves in the one game can be mimicked with moves in the other game and vice versa, until either both matches reach a literal, in which case the winner is the same, or the real match reaches a position (t, a) for some $a \in A$. If $a \notin C$ then the shadow match is at (t, ξ_a) , and we can use the induction hypothesis to get $(\mathbb{A}, a) \equiv \xi_a$, hence both positions are winning for \exists . If $a \in C$ then $a = c_j$ for some $j \in \{1, \dots, n\}$. If $j \in I$ then the shadow match is at $(t, x_{I(j)})$ and necessarily continues to $(t, \varphi_{I(j)})$. Because $j = \min(I(j))$ we can take $J := I(j)$ to get that the real match is at (t, c_J) and the shadow match is at (t, φ_J) . If $j \notin I$ then the shadow match is at $(t, \varphi_{I(j) \cup \{j\}})$. Now taking $J := I(j) \cup \{j\}$ gets us that the real match is at (t, c_J) and the shadow match is at (t, φ_J) .

It remains to check that the winner is the same if both matches are infinite. In this case there is a \preceq -greatest variable x_M that occurs infinitely often during the shadow match. Since every position (t, x_M) is followed by (t, φ_M) , c_M is visited infinitely often in the real match. Since X is finite, there is a point after which the only variables that are visited in the shadow match are variables that are visited infinitely often. Let ψ be a formula that occurs in the trace induced by the shadow match after this point, then

walking backwards through the trace we find only superformulas of ψ , until we go from a superformula $\lambda_I x_I. \vartheta_I$ to x_I . Now $x_I \preceq x_M$ hence φ_I is a subformula of φ_M , which means also ψ is a subformula of φ_M . As a result every φ_I that occurs infinitely often in the shadow match is a subformula of φ_M , and this means $r(I) \leq r(M)$. Take $m := \min(M)$ then we show that $I(m) = M$ for all φ_I that occur infinitely often; certainly $M(m) = M$. If φ_I with $I(m) = M$ is followed by φ_J where $J := I(j) \cup \{j\}$ for some $j \notin I$, then $j > m$ would imply $I(j) = \{i \in I \mid i > j > m\} \subseteq M \setminus \{m\}$ hence $J \subsetneq M$ which contradicts $r(J) \leq r(M)$. Therefore $j \leq m$ hence $M \subseteq I(j)$ which gives $J(m) = M$. This means that $m \in I$ for all φ_I that occur infinitely often in the shadow match. For every state c_I that occurs infinitely often in the real match we have that φ_I occurs infinitely often in the shadow match, but then $\min(I) \leq m = \min(M)$. Since the numbering of C is from low to high priority, this means $\Omega(c_I) \leq \Omega(c_M)$. Therefore $\Omega(c_M)$ is the highest priority among states visited infinitely often in the real match. Because the shadow match is consistent with f it is won by \exists , which means that $\lambda_M = \nu$ and so $\Omega(c_M)$ must be even. We conclude that \exists wins the real match.

For the converse, let \mathbb{S} be a transition system, let $s_I \in S$ and suppose that $\mathbb{S}, s_I \not\models \varphi_{\{k\}}$. By Proposition 2.29 this means \forall has a winning strategy. The rest of the proof is dual. \square

5.2 Dominated fixpoints

A nice property of dominance is that it applies both to modal formulas and to modal automata; the following definition generalizes Definition 3.15 to arbitrary μML formulas.

Definition 5.10. A $\mu\text{ML}(Y)$ formula ξ is *X-free* for some $X \subseteq Y$ if $\text{FV}(\xi) \subseteq Y \setminus X$. It is *\exists -dominated* with respect to X if every subformula $\Box\varphi$ is X -free and for every subformula $\varphi \wedge \psi$ either φ or ψ is X -free. It is *\forall -dominated* with respect to X if every subformula $\Diamond\varphi$ is X -free and for every subformula $\varphi \vee \psi$ either φ or ψ is X -free. It is *dominated* with respect to X if it is either \exists -dominated or \forall -dominated. \triangleleft

In fact, the class of modal automata that have dominated clusters can be characterized by a fragment of the modal μ -calculus, which we will call $\mu_{\text{dom}}\text{ML}$.

Definition 5.11. A μML formula has *dominated fixpoints* if for every subformula of the form $\lambda x. \vartheta$ the formula ϑ is either \exists -dominated or \forall -dominated with respect to $\text{FV}(\vartheta)$. The fragment $\mu_{\text{dom}}\text{ML}$ of $\mu_1\text{ML}$ consists of all formulas that have dominated fixpoints. \triangleleft

Note that here we take $\mu_{\text{dom}}\text{ML}$ as a fragment of $\mu_1\text{ML}$, so that we can apply the automata constructions from section 5.1. Because dominance is a rather syntactic property, it is preserved under these constructions; we will see that for every $\mu_{\text{dom}}\text{ML}$ formula ξ the automaton \mathbb{A}_ξ has dominated clusters, and for every automaton \mathbb{A} with dominated clusters the formula $\xi_{\mathbb{A}}$ is in $\mu_{\text{dom}}\text{ML}$.

The direction from automata to formulas is relatively easy: the fixpoint formulas that correspond to states in a \exists -dominated cluster C are \exists -dominated over their free variables, because these free variables also correspond to states in C .

Proposition 5.12. *If \mathbb{A} has dominated clusters then $\xi_{\mathbb{A}}$ has dominated fixpoints.*

Proof. Let φ be a fixpoint subformula of $\xi_{\mathbb{A}}$, then there is a cluster C and a set $J \in \mathcal{I}$ such that $\varphi = \varphi_J = \lambda_J x_J. \vartheta_J$. The cluster C is either \exists -dominated or \forall -dominated. We assume the former; the latter case is dual. We will show using induction on the rank r that ϑ_I is \exists -dominated with respect to $\text{FV}(\vartheta_I)$ for all $I \in \mathcal{I}$, so that in particular ϑ_J is \exists -dominated with respect to $\text{FV}(\vartheta_J)$.

Let $I \in \mathcal{I}$. For a formula $\alpha \in \text{gML}_1(A)$, let α' be created from α by replacing every $a \in A \setminus C$ with ξ_a , every $c_i \in C$ where $i \in I$ with $x_{I(i)}$ and every $c_j \in C$ where $j \notin I$ with $\varphi_{I(j) \cup \{j\}}$, so that $\vartheta_I = \Delta(c_I)'$. It may be clear that subformulas of ϑ_I are either of the form α' where α is a subformula of $\Delta(c_I)$, or subformulas of $\varphi_{I(j) \cup \{j\}}$ for some $c_j \in C$ with $j \notin I$. The induction hypothesis states that for all $c_j \in C$ with $j \notin I$ that occur in $\Delta(c_I)$, $\vartheta_{I(j) \cup \{j\}}$ is \exists -dominated with respect to $\text{FV}(\vartheta_{I(j) \cup \{j\}})$. If a variable $x \in \text{FV}(\vartheta_I)$ were to occur in $\vartheta_{I(j) \cup \{j\}}$ then it would be in $\text{FV}(\vartheta_{I(j) \cup \{j\}})$. Therefore we only need to check subformulas of the former kind.

Let $\Box\beta$ be a subformula of $\Delta(c_I)$ then $\beta = a$ for some $a \in A \setminus C$ because $\Delta(c_I)$ is \exists -dominated with respect to C . This means that $\beta' = \xi_a$ and $\text{FV}(\xi_a) = \emptyset$, which means that $\Box\beta'$ cannot contain occurrences of variables from $\text{FV}(\vartheta_I)$. Similarly if $\beta_1 \wedge \beta_2$ is a subformula of $\Delta(c_I)$, then one of β_1 and β_2 is in $\text{gML}_1(A \setminus C)$, whence either $\text{FV}(\beta'_1) = \emptyset$ or $\text{FV}(\beta'_2) = \emptyset$ and therefore one of β'_1 and β'_2 does not contain any occurrences of variables from $\text{FV}(\vartheta_I)$. \square

For the other direction, we need two more propositions about relations \triangleleft_{ξ} and \triangleright_{ξ} .

Proposition 5.13. *Let ξ be a μML formula. For every non-trivial \triangleright_{ξ} equivalence class Φ there is $x \in \text{BV}(\xi)$ such that $x \in \Phi$, $\lambda_x x. \vartheta_x \in \Phi$ and $\Phi \subseteq \text{Sb}(\lambda_x x. \vartheta_x)$.*

Proof. Since $\Phi \subseteq \text{Sb}(\xi)$ we know that Φ is finite, say $\Phi = \{\varphi_0, \dots, \varphi_n\}$. By definition of \triangleright_{ξ} there are $\psi_{i,0}, \dots, \psi_{i,k_i}$ such that $\psi_{i,0} = \varphi_i$ for all $i \leq n$, $\psi_{i,k_i} = \varphi_{i+1}$ for all $i < n$, $\psi_{n,k_n} = \varphi_0$ and $\psi_{i,j+1} \in \nabla_{\xi}(\psi_{i,j})$ for all $i \leq n$ and $j < k_i$. Now we can construct a direct ξ -trace by repeating $\psi_{0,0}, \dots, \psi_{0,k_0-1}, \psi_{1,0}, \dots, \psi_{n,k_n-1}$. Note that $\varphi_0 \triangleright_{\xi} \psi_{i,j} \triangleright_{\xi} \varphi_0$ and thus $\psi_{i,j} \in \Phi$ for all i and j , and that every φ_i occurs infinitely often in the trace. By Proposition 2.25 we get a \preceq -greatest bound variable x of ξ that occurs infinitely often during the trace. Because x will always be followed by $\lambda_x x. \vartheta_x$ we have $\lambda_x x. \vartheta_x \in \Phi$. Now if we start at φ_i and go backwards through the trace, then we will always get a superformula of φ_i until we go from $\lambda_y y. \vartheta_y$ to y for some bound variable y . Note that φ_i is a subformula of $\lambda_y y. \vartheta_y$. For this y we have $y \preceq x$, which means $\lambda_y y. \vartheta_y$ is a subformula of $\lambda_x x. \vartheta_x$. Therefore φ_i is a subformula of $\lambda_x x. \vartheta_x$, and so $\Phi \subseteq \text{Sb}(\lambda_x x. \vartheta_x)$. \square

Next, we note that bound variables are the only μML -formulas that have derivatives which are not subformulas. As a result, the only way to go from a formula φ to something which isn't a subformula of φ , is by visiting a variable that is bound by some superformula of φ . For this to be possible, that variable must be free in φ .

Proposition 5.14. *Let ξ be a μML formula and let $\varphi, \psi \in \text{Sb}(\xi)$. If $\text{FV}(\varphi) = \emptyset$ then $\varphi \triangleright_{\xi} \psi$ iff $\psi \in \text{Sb}(\varphi)$.*

Proof. Let ξ and φ be as such. Of course $\varphi \triangleright_\xi \psi$ for all $\psi \in \text{Sb}(\varphi)$. Let Φ be the closure of $\text{Sb}(\varphi)$ under \triangleright_ξ , then we can construct Φ as $\bigcup_{n \in \mathbb{N}} \Phi_n$ where $\Phi_0 = \text{Sb}(\varphi)$ and where Φ_{n+1} consists of all the ξ -derivatives of formulas in Φ_n . We have $\Phi_0 \subseteq \text{Sb}(\varphi)$. Suppose $\Phi_n \subseteq \text{Sb}(\varphi)$ and let $\psi \in \nabla_\xi(\psi')$ for some $\psi' \in \Phi_n$. Either ψ is a subformula of $\psi' \in \text{Sb}(\varphi)$, in which case $\psi \in \text{Sb}(\varphi)$, or $\psi' = x$ for some $x \in \text{BV}(\xi)$ and $\psi = \lambda_x x. \vartheta_x$. Because x is a subformula of φ and φ has no free variables, x must be bound by some subformula of φ . Since ξ is clean this formula must be ψ , which means $\psi \in \text{Sb}(\varphi)$. This shows $\Phi_{n+1} \subseteq \text{Sb}(\varphi)$ and by induction $\Phi \subseteq \text{Sb}(\varphi)$. Now if $\psi \in \text{Sb}(\xi)$ is such that $\varphi \triangleright_\xi \psi$, then $\psi \triangleleft_\xi \varphi \in \text{Sb}(\varphi)$ hence $\psi \in \Phi \subseteq \text{Sb}(\varphi)$. \square

Now the other direction can be proven. The trick lies in identifying which fixpoint formulas of ξ belong to the same \triangleleft_ξ class, because Proposition 5.6 then gives that the corresponding states form a cluster of \mathbb{A}_ξ .

Proposition 5.15. *If $\xi \in \mu_1 \text{ML}_\Lambda$ has dominated fixpoints then \mathbb{A}_ξ has dominated clusters.*

Proof. Let ξ be a $\mu_{\text{dom}} \text{ML}_\Lambda$ formula. Let C be a non-trivial cluster of \mathbb{A}_ξ and define $Y := \{x \in \text{BV}(\xi) \mid \underline{x} \in C\}$. By Proposition 5.6, all $\lambda(x)$ for $x \in Y$ belong to the same non-trivial \triangleleft_ξ class, and no other fixpoint formulas do. By Proposition 5.13 there is a $y \in Y$ with $\lambda(x) \in \text{Sb}(\lambda(y))$ for all $x \in Y$. Now ϑ_y is either \exists -dominated or \forall -dominated with respect to $\text{FV}(\vartheta_y)$. We assume that it is \exists -dominated; the case where it is \forall -dominated is dual. We will show that $\Delta_\xi(\underline{x})$ is \exists -dominated with respect to C for all $x \in Y$, so that C is an \exists -dominated cluster.

If we write $x \mapsto x'$ whenever $\lambda(x')$ occurs in the decomposition of ϑ_x , then every $x \in Y$ is reachable by a finite number of \mapsto steps from y . Therefore we can use finite induction on \mapsto , starting with y . We assumed that ϑ_y is \exists -dominated with respect to $\text{FV}(\vartheta_y)$. Now if $x \in Y$ is such that ϑ_x is \exists -dominated with respect to $\text{FV}(\vartheta_x)$, then we will show that $\Delta_\xi(\underline{x})$ is \exists -dominated with respect to C and that if $x \mapsto x'$ then ϑ'_x is \exists -dominated with respect to $\text{FV}(\vartheta_{x'})$.

Let $x \in Y$ be such that ϑ_x is \exists -dominated with respect to $\text{FV}(\vartheta_x)$. Let ϑ_x be decomposed as $\alpha(z_1, \dots, z_j, \lambda(z_{j+1}), \dots, \lambda(z_k))$ where $\alpha \in \text{ML}(\mathbb{N})$ and $z_1, \dots, z_k \in \text{BV}(\xi)$, so that $\Delta_\xi(\underline{x}) = \alpha(\underline{z}_1, \dots, \underline{z}_k)$, $z_1, \dots, z_j \in \text{FV}(\vartheta_x)$ and $\text{FV}(\vartheta_{z_i}) \subseteq \text{FV}(\vartheta_x) \cup \{z_i\}$ for $i > j$. It may be clear that the subformulas of $\Delta_\xi(\underline{x})$ are the form $\beta(\underline{z}_1, \dots, \underline{z}_k)$ where $\beta \in \text{ML}(\mathbb{N})$ is a subformula of α , and that $\beta(z_1, \dots, z_j, \lambda(z_{j+1}), \dots, \lambda(z_k))$ is then a subformula of ϑ .

Let $\square\beta$ be a subformula of α , then because ϑ_x is \exists -dominated $\beta(z_1, \dots, \lambda(z_k))$ does not contain occurrences from variables in $\text{FV}(\vartheta_x)$, which means that $\beta(\underline{z}_1, \dots, \underline{z}_k)$ does not contain occurrences of $\underline{z}_1, \dots, \underline{z}_j$. Let $i > j$ such that \underline{z}_i occurs in $\beta(\underline{z}_1, \dots, \underline{z}_k)$, then $\lambda(z_i)$ occurs in $\beta(z_1, \dots, \lambda(z_k))$ and therefore does not contain occurrences from variables in $\text{FV}(\vartheta_x)$, hence $\text{FV}(\lambda(z_i)) = \emptyset$. Because β is a proper subformula of $\square\beta$, $\lambda(z_i)$ is a proper subformula of $\lambda(x)$. Now by Proposition 5.14 and Proposition 5.5 this means $\underline{z}_i \not\mapsto \underline{x}$ and thus $\underline{z}_i \notin C$. Therefore $\beta(\underline{z}_1, \dots, \underline{z}_n)$ does not contain any occurrences of states from C .

Let $\beta_1 \wedge \beta_2$ be a subformula of α , then because ϑ is \exists -dominated either $\beta_1(z_1, \dots, \lambda(z_k))$ or $\beta_2(z_1, \dots, \lambda(z_k))$ does not contain occurrences from variables in $\text{FV}(\vartheta_x)$. Without loss

of generality we assume it is β_1 . As above we find that $\beta_1(\underline{z}_1, \dots, \underline{z}_n)$ does not contain any occurrences of states from C .

Let $x' \in Y$ such that $x \rightsquigarrow x'$, thus $\lambda(x')$ occurs in the decomposition of ϑ_x . Note that $\text{FV}(\lambda(x')) \subseteq \text{FV}(\vartheta_x)$. It cannot be the case that $\text{FV}(\lambda(x')) = \emptyset$, because then Proposition 5.14 and Proposition 5.5 tell us that $\underline{x}' \not\rightsquigarrow \underline{x}$ whence $\underline{x}' \notin C$. Thus there is some $z \in \text{FV}(\vartheta_x)$ that occurs in $\lambda(x')$ and hence in $\vartheta_{x'}$. Because $\vartheta_{x'}$ is a μ_1 ML formula, this z occurs in some subformula of $\vartheta_{x'}$ that is either of the form $\diamond\psi$ or of the form $\square\psi$. But this subformula is also a subformula of ϑ_x , so it must be $\diamond\psi$. This means that $\vartheta_{x'}$ cannot be \forall -dominated with respect to $\text{FV}(\vartheta_{x'})$. Because ξ has dominated fixpoints, it must be \exists -dominated. \square

5.3 Mostowski index and alternation depth

Next we will look at modifications to the priority map of an automaton. Armed with Proposition 3.21, we can obtain an interesting result about the Mostowski indices of modal parity automata with dominated clusters. For arbitrary parity automata, there is a strict hierarchy based on the Mostowski index. For guarded modal parity automata with dominated clusters, this hierarchy collapses to the class of automata whose \exists -dominated clusters are Büchi, i.e. have a Mostowski index inside $(1, 2)$, and whose \forall -dominated clusters are co-Büchi, i.e. inside $(0, 1)$.

First, Proposition 2.50 tells us that we can make sure that the dominated clusters have Mostowski indices of the form $(1, h)$ and $(0, h)$ respectively.

Definition 5.16. Let $\mathbb{A} = (A, \Delta, \Omega, a_I)$ be a modal parity automaton with dominated clusters. Let C be a cluster of \mathbb{A} and let (l, h) be its Mostowski index. Let $c \in C$. If l is odd, define $\Omega^\downarrow(c) := \Omega(c) - l + 1$. If l is even and C is \exists -dominated, define $\Omega^\downarrow(c) := \Omega(c) - l + 2$. If l is even and C is \forall -dominated, define $\Omega^\downarrow(c) := \Omega(c) - l$. Define $\mathbb{A}^\downarrow = (A, \Delta, \Omega^\downarrow, a_I)$. \triangleleft

Proposition 5.17. *For every guarded modal parity automaton \mathbb{A} with dominated clusters, the guarded modal parity automaton \mathbb{A}^\downarrow is equivalent to \mathbb{A} and has dominated clusters.*

Proof. Since the transition terms remain unchanged, \mathbb{A}^\downarrow is a guarded modal parity automaton with dominated clusters. Equivalence follows from Proposition 2.50. \square

The key then lies in bringing down the height of the highest priorities that are needed inside a cluster. Proposition 3.21 tells us that in an \exists -dominated cluster, \exists can predict which states the acceptance match will visit for as long as the acceptance match remains in the cluster. This gives her two alternatives: either the match visits some states with priorities deemed “high” infinitely often, in which case it is irrelevant which states with lower priorities the match will visit, or there comes a point after which the match only visits states with priorities deemed “low”, in which case she can commit to never visiting a state with a high priority again.

To make this commitment explicit, we can divide every \exists -dominated cluster into a low part and a high part. For every cluster a copy of the low part is added. The states in the

original cluster will all have high priorities, and the states in the copy will all have low priorities. At a state in the original, \exists can now decide to either stay in the original cluster or to go to the copy. Once inside the copy, \exists cannot go back; hence her commitment.

The following definition simultaneously divides all the clusters.

Definition 5.18. Let $\mathbb{A} = (A, \Delta, \Omega, a_1)$ be a guarded modal parity automaton with dominated clusters. Create for every $a \in A$ a new state \bar{a} . For $X \subseteq A$ write $\bar{X} := \{\bar{a} \mid a \in X\}$. Let C be a cluster of \mathbb{A} . Take $m := 2$ if C is \exists -dominated and $m := 1$ if it is \forall -dominated. Divide C into $C_0 := \{c \in C \mid \Omega(c) \leq m\}$ and $C_1 := \{c \in C \mid \Omega(c) > m\}$. For a formula $\varphi \in \text{gML}_1(A)$ create φ^- from φ by replacing every occurrence of c for $c \in C_0$ with \bar{c} , every occurrence of $\diamond c$ for $c \in C_1$ with \perp and every occurrence of $\square c$ for $c \in C_1$ with \top . For $c \in C$, define $\Delta^\dagger(c) := \Delta(c) \vee \Delta(c)^-$, $\Delta^\dagger(\bar{c}) = \Delta(c)^-$, $\Omega^\dagger(c) := \max\{\Omega(c), m + 1\}$ and $\Omega^\dagger(\bar{c}) := \min\{\Omega(c), m\}$. Define $\mathbb{A}^\dagger := (A \cup \bar{A}, \Delta^\dagger, \Omega^\dagger, a_1)$. \triangleleft

Note that the threshold m and the substitution \cdot^- on $\text{gML}_1(A)$ are relative to the cluster C , but because we will only look at one cluster at a time this is not reflected in the notation. Also note that $\Omega(c) = \Omega^\dagger(c)$ for all $c \in C_1$, and $\Omega(c) = \Omega^\dagger(\bar{c})$ for all $c \in C_0$. The clusters of \mathbb{A} are replaced by generalized clusters of \mathbb{A}^\dagger , and dominance is preserved.

Proposition 5.19. *For every guarded modal parity automaton \mathbb{A} with dominated clusters, the guarded modal parity automaton \mathbb{A}^\dagger has dominated clusters.*

Proof. Let C be an \exists -dominated cluster of \mathbb{A} , then $C \cup \bar{C}$ is a generalized cluster of \mathbb{A}^\dagger . Both $\Delta^\dagger(c)$ and $\Delta^\dagger(\bar{c})$ are \exists -dominated over $C \cup \bar{C}$, and thus $C \cup \bar{C}$ is \exists -dominated. Similarly for every \forall -dominated cluster C of \mathbb{A} , $C \cup \bar{C}$ is an \forall -dominated generalized cluster of \mathbb{A}^\dagger . Now the \mathbb{A}^\dagger -clusters of states in $C \cup \bar{C}$ are subsets of $C \cup \bar{C}$, hence will also be dominated. \square

For a cluster C of \mathbb{A} we write $C' := C \cup \bar{C}_0$. Note that C' is a generalized cluster of \mathbb{A}^\dagger . In order to prove that \mathbb{A}^\dagger is equivalent to \mathbb{A} , we will show that the states in a cluster C are equivalent to their counterparts in C' . For the following two propositions, we use the cluster-path acceptance games for the cluster C of \mathbb{A} and the generalized cluster C' of \mathbb{A}^\dagger .

Proposition 5.20. *Let \mathbb{A} be a guarded modal parity automaton. Let C be a cluster of \mathbb{A} and let $a \in C$. Suppose that $(\mathbb{A}, b) \equiv (\mathbb{A}^\dagger, b)$ for all $b \in A$ of cluster-depth less than that of a . Let \mathbb{S} be a transition system and let $\pi \in \Pi(\mathbb{S})$. If a player has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@ (0, a)$, then that player has a winning strategy for $\mathcal{A}(\mathbb{A}^\dagger, \mathbb{S}, C', \pi)@ (0, a)$.*

Proof. Suppose the \exists has a winning strategy; the case where \forall has a winning strategy is dual. Let f be her strategy. By Proposition 3.21 there is a function $\epsilon : \mathbb{N} \rightarrow C$ such that any position (i, c) with $c \in C$ in a match consistent with f has $c = \epsilon(i)$. If there is a least k such that $\Omega(\epsilon(i)) \geq m$ for all $i \geq k$, then define $\epsilon' : \mathbb{N} \rightarrow C'$ by $\epsilon'(i) := \epsilon(i)$ for $i < k$ and $\epsilon'(i) := \bar{\epsilon}(i)$ for $i \geq k$. If not then for every k there is $i \geq k$ with $\Omega(\epsilon(i)) > m$; in that case define $\epsilon' := \epsilon$. Now note that $\epsilon'(k) \notin C$ iff $\Omega(\epsilon(i)) \leq m$ for all $i \geq k$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}^\dagger, \mathbb{S}, C', \pi)@ (0, a)$, \exists plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@ (0, a)$ that is consistent with f .

Suppose both matches are at a position (i, c) with $c \in C$. The real match necessarily continues to $(i, \Delta^\dagger(c)) = (i, \Delta(c) \vee \Delta(c)^-)$ and the shadow match continues to $(i, \Delta(c))$. If $\epsilon'(i+1) \in C$ then \exists should move to $(i, \Delta(c))$. Otherwise she should move to $(i, \Delta(c)^-)$. Suppose the shadow match is at (i, c) with $c \in C_0$ and the real match is at (i, \bar{c}) , then the matches necessarily continue to $(i, \Delta(c))$ and $(i, \Delta(c)^-)$ respectively. Suppose both matches are at (i, φ) with $\varphi \in \text{gML}_1(A)$, then the two matches continue identically. Suppose that the shadow match is at (i, φ) with $\varphi \in \text{gML}_1(A)$, that the real match is at (i, φ^-) and that $\epsilon'(i+1) \notin C$. The two matches continue virtually identically with real moves to (i, ψ^-) corresponding to shadow moves to (i, ψ) and vice versa. The shadow match will not reach a position $(i, \diamond c)$ with $c \in C_1$ because then $c = \epsilon(i+1)$ which contradicts $\Omega(\epsilon(i+1)) \leq m$. If the shadow match is at $(i, \diamond b)$ or $(i, \square b)$ with $b \notin C$, then the real match is at that position as well. Clearly any move to (t, b) in one match can be mimicked in the other match, and from there the winners of both matches are the same because $(\mathbb{A}, b) \equiv (\mathbb{A}^\dagger, b)$.

In this way the two matches are played to completion. If both matches are infinite, then the states visited in the shadow match are the states of ϵ , whereas the states visited in the real match are the states of ϵ' . Because the shadow match is consistent with f hence won by \exists , the highest priority among its states must be even. Let x be this priority. If $x > m$ then there are infinitely many states in ϵ with priority bigger $x > m$, hence $\epsilon' = \epsilon$. The only states $c \in C$ for which $\Omega^\dagger(c) \neq \Omega(c)$ are those where $\Omega(c) < m+1$, but then $\Omega(c) < x$ as well. This means that the highest priority among states visited in the real match is also x , hence \exists wins. If $x \leq m$ then there is k with $\Omega(\epsilon(i)) \leq m$ for all $i \geq k$, but then $\epsilon'(i) \in \bar{C}_0$ hence $\Omega^\dagger(\epsilon'(i)) = \Omega(\epsilon(i))$ for all $i \geq k$. Thus the highest priority occurring infinitely often in the real match is x . We conclude that \exists wins the real match. \square

Proposition 5.21. *Let \mathbb{A} be a guarded modal parity automaton. Let C be a cluster of \mathbb{A} and let $a \in C$. Suppose that $(\mathbb{A}, b) \equiv (\mathbb{A}^\dagger, b)$ for all $b \in A$ of cluster-depth less than that of a . Let \mathbb{S} be a transition system and let $\pi \in \Pi(\mathbb{S})$. If \exists has a winning strategy for $\mathcal{A}(\mathbb{A}^\dagger, \mathbb{S}, C', \pi)@(0, a)$, then she has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$.*

Proof. Let g be her strategy. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$, \exists plays a shadow match of $\mathcal{A}(\mathbb{A}^\dagger, \mathbb{S}, C', \pi)@(0, a)$ that is consistent with g .

Suppose the two matches are at position (i, c) with $c \in C$. The real match necessarily continues to $(i, \Delta(c))$ and the shadow match continues to $(i, \Delta^\dagger(c)) = (i, \Delta(c) \vee \Delta(c)^-)$. There g instructs \exists to move to either $(i, \Delta(c))$ or $(i, \Delta(c)^-)$. Suppose the real match is at (i, c) with $c \in C_0$ and the shadow match is at (i, \bar{c}) , then the matches necessarily continue to $(i, \Delta(c))$ and $(i, \Delta(c)^-)$ respectively. Suppose the two matches are at (i, φ) with $\varphi \in \text{gML}_1(A)$, then the two matches continue identically. Suppose the real match is at (i, φ) with $\varphi \in \text{gML}_1(A)$ and that the shadow match is at (i, φ^-) . The two matches continue virtually identically with real moves to (i, ψ) corresponding to shadow moves to (i, ψ^-) and vice versa. The shadow match will not reach a position $(i, (\diamond c)^-)$ with $c \in C_1$ since $(\diamond c)^- = \perp$ and (i, \perp) is losing for \exists . If the shadow match is at $(i, \diamond b)$ or $(i, \square b)$ with $b \notin C$, then the real match is at that position as well. Clearly any move to (t, b) in one match can be mimicked in the other match, and from there the winners of both matches are the same because $(\mathbb{A}, b) \equiv (\mathbb{A}^\dagger, b)$.

In this way the two matches are played to completion. If both matches are infinite, then a position (i, c) with $c \in C$ in the shadow match corresponds to that same position in the real match and a position (i, \bar{c}) with $c \in C_0$ in the shadow match corresponds to (i, c) in the real match. Because the shadow match is consistent with g hence won by \exists , the highest priority among states visited in the shadow match must be some even x . If $x > m$ then $x > m + 1$ because $m + 1$ is odd, and thus $\Omega(c) = \Omega^\dagger(c) = x$ for infinitely many $c \in C$. Therefore the highest priority among states in the real match must also be x . If $x \leq m$ then after a certain point only states \bar{c} are visited in the shadow match, but that then means that in the real match positions c with $c \in C_0$ hence $\Omega(c) = \Omega^\dagger(\bar{c})$ are visited. Thus the highest priority among states in the real match must again be x . We conclude that \exists wins the real match. \square

With these two propositions in place, we can apply induction to get $\mathbb{A} \equiv \mathbb{A}^\dagger$.

Proposition 5.22. *For every guarded modal parity automaton \mathbb{A} with dominated clusters, the guarded modal parity automaton \mathbb{A}^\dagger is equivalent to \mathbb{A} .*

Proof. We prove $(\mathbb{A}, a) \equiv (\mathbb{A}^\dagger, a)$ for all $a \in A$ by induction on the cluster depth of a , so that in particular $\mathbb{A} \equiv \mathbb{A}^\dagger$. Let C be a cluster of \mathbb{A} and let $c \in C$. The induction hypothesis states that $(\mathbb{A}, b) \equiv (\mathbb{A}^\dagger, b)$ for all b of cluster-depth less than that of c . In light of Proposition 2.43 and Proposition 3.20, Proposition 5.20 and Proposition 5.21 prove that $(\mathbb{A}, c) \equiv (\mathbb{A}^\dagger, c)$. \square

The above construction does not actually decrease the Mostowski index of the automaton. However, if the automaton had an \exists -dominated cluster with index $(1, 6)$, then this has been replaced with \exists -dominated clusters that have indices inside $(1, 2)$ and inside $(3, 6)$. These can then be shifted down to get clusters with indices inside $(1, 2)$ and $(1, 4)$. Repeating this process enough times results in Büchi and co-Büchi clusters.

Theorem 5.23. *For every guarded modal parity automaton with dominated clusters there is an equivalent guarded modal parity automaton with dominated clusters whose \exists -dominated clusters are Büchi and whose \forall -dominated clusters are co-Büchi.*

Proof. Let \mathbb{A} be a guarded modal automaton with dominated clusters. By Proposition 5.17, \mathbb{A}^\dagger is equivalent to \mathbb{A} and has dominated clusters. The \exists -dominated clusters of \mathbb{A}^\dagger have a Mostowski index of (l, h) where $l \in \{1, 2\}$, and the \forall -dominated clusters have a Mostowski index of (l, h) where $l \in \{0, 1\}$. By repeated application of the modifications \cdot^\dagger and \cdot^\downarrow , the Mostowski indices of \exists -dominated clusters can be brought down to $(1, 2)$, while the Mostowski indices of \forall -dominated clusters are brought down to $(0, 1)$. By Proposition 5.19, Proposition 5.22 and Proposition 5.17, the resulting automaton is equivalent to \mathbb{A} . \square

Low Mostowski indices correspond with low alternation depth. The above theorem states that the Mostowski index hierarchy collapses for automata with dominated clusters. When this result is combined with the theorems from section 5.2, we obtain a similar collapse in the alternation depth hierarchy for $\mu_{\text{dom}}\text{ML}$.

Theorem 5.24. *For every μ_{dom} ML formula there exists an equivalent guarded modal parity automaton with dominated clusters whose \exists -dominated clusters are Büchi and whose \forall -dominated clusters are co-Büchi.*

Proof. An immediate result of Theorem 5.7, Proposition 5.15 and Theorem 5.23. \square

Theorem 5.25. *For every guarded modal parity automaton with dominated clusters there exists an equivalent μ_{dom} ML formula of alternation depth at most 2.*

Proof. This follows from Theorem 5.23, Theorem 5.9 and Proposition 5.12. Note that if \mathbb{A} is a guarded modal parity automaton with dominated clusters whose \exists -dominated clusters are Büchi and whose \forall -dominated clusters are co-Büchi, then the formula $\xi_{\mathbb{A}}$ will have alternation depth at most 2. \square

Theorem 5.26. *For every μ_{dom} ML formula there exists an equivalent μ_{dom} ML formula of alternation depth at most 2.*

Proof. A direct result of Theorem 5.24 and Theorem 5.25. \square

Chapter 6

A characterization of CTL

When CTL was introduced by Clarke and Emerson (1981), they immediately gave an interpretation of the CTL operators as fixpoints of one-variable maps. It is indeed relatively easy to come up with an inductive translation of CTL into the modal μ -calculus using only one variable. In this chapter, we will give an exact characterization of the expressive power of CTL, both in terms of a class of modal automata and as a fragment of μ ML.

6.1 Automata for CTL

In chapter 4 we saw that for every CTL* formula there is an equivalent guarded modal parity automaton with dominated clusters. In this section we will see that CTL can be characterized by restricting this class of automata to those automata that have singleton clusters. We start with showing that for a CTL_Σ formula ξ the clusters of the automaton $(\mathbb{A}_\xi \odot \mathbb{B}_\xi)^\gamma$ are of size 1. The key observation is that if ξ is in CTL then the Dam terms used to construct \mathbb{A}_ξ contain at most one actual path-formula, and this formula is relatively simple. First, let us define a measure of complexity that can formalize this with.

Definition 6.1. Define the *explosive complexity* of a CTL* formula φ , $\text{xc}(\varphi)$, by

$$\begin{aligned} \text{xc}(\ell) &:= 0 \\ \text{xc}(\varphi \star \psi) &:= 5 * \text{xc}(\varphi) + 5 * \text{xc}(\psi) && (\star \in \{\vee, \wedge\}) \\ \text{xc}(Q\varphi) &:= 0 && (Q \in \{E, A\}) \\ \text{xc}(X\varphi) &:= \text{xc}(\varphi) + 2 \\ \text{xc}(\varphi \star \psi) &:= 5 * \text{xc}(\varphi) + 5 * \text{xc}(\psi) + 1 && (\star \in \{U, R\}) \end{aligned}$$

where $\ell \in \text{Lit}$. ◁

The number 5 here is arbitrary, but chosen large enough so that non-zero values are blown up. In this way, if $\text{xc}(\varphi) > 0$ or $\text{xc}(\psi) > 0$ then $\text{xc}(\varphi \wedge \psi) \geq 5$. A simple inductive proof shows that $\text{xc}(\varphi) = 0$ for every $\varphi \in \text{CTL}_\Sigma^*$. Conversely if φ is an actual path-formula, i.e. $\varphi \in \text{CTL}_\Pi^* \setminus \text{CTL}_\Sigma^*$, then $\text{xc}(\varphi) > 0$. Lastly $\text{CTL}_\Pi \setminus \text{CTL}_\Sigma$ formulas are formulas of the

form $X\alpha$, $\alpha U\beta$ and $\alpha R\beta$ where $\alpha, \beta \in \text{CTL}_\Sigma$, and this means $\text{xc}(\varphi) \leq 2$ for every $\varphi \in \text{CTL}$. Again, it is not hard to show that if $\text{xc}(\varphi) > 2$ then $\varphi \notin \text{CTL}$.

In this way xc can identify formulas that are small enough to be in CTL. However in section 3.1 we noted that CTL is not closed under derivatives; the following proposition ensure that formulas in $\text{Cl}(\text{CTL})$ are still small in terms of xc .

Proposition 6.2. *For every $\varphi \in \text{Cl}(\text{CTL})$, $\text{xc}(\varphi) \leq 3$.*

Proof. Let $\varphi \in \text{Cl}(\text{CTL})$, then by Proposition 3.10 there is a CTL formula ψ such that $\varphi \bowtie \psi$. Now Proposition 3.6 tells us that either $\varphi = \psi$, in which case $\text{xc}(\varphi) = \text{xc}(\psi) \leq 2$, or φ and ψ both belong to the \bowtie -class of some χ , where χ is either a U-formula or an R-formula. Looking at the formulas in this cell it becomes clear that only χ can be in CTL, hence $\chi = \psi$ and $\text{xc}(\psi) = 1$, and that $\text{xc}(\varphi) \leq \text{xc}(\chi) + 2$. Thus $\text{xc}(\varphi) \leq 3$. \square

Now we can count the number of path-formulas that occur in a state of the automaton \mathbb{A}_ξ . If ξ is a CTL_Σ formula, we obtain that there is at most one path-formula per state.

Definition 6.3. Let ξ be a CTL_Σ^* formula. Define the *path-formula count* of a state $QZ \in A_\xi$, $\text{pc}(QZ)$, by $\text{pc}(QZ) := \#\{\varphi \in \text{Ran}(Z) \mid \text{xc}(\varphi) > 0\}$. \triangleleft

Proposition 6.4. *Let ξ be a CTL_Σ formula. For any reachable state $a \in A_\xi$, $\text{pc}(a) \leq 1$.*

Proof. It certainly holds for Q_1Z_1 since Z_1 is a singleton. Let QZ be reachable and non-terminal, and suppose that $\text{pc}(QZ) \leq 1$. Let $\tau : QZ \Rightarrow Q_1Z_1, \dots, Q_nZ_n$ be the applicable rule for QZ . Let $\Phi := \text{Ran}(Z)$ and $\Phi_i := \text{Ran}(Z_i)$.

If it is an extraction rule then there is a formula $\varphi \in \Phi$ that is being extracted. Note that $\varphi \in \text{CTL}_\Sigma$ hence $\text{xc}(\varphi) = 0$. Now $\Phi_1 = \Phi \setminus \{\varphi\}$ and $\Phi_2 = \{\varphi\}$, and therefore $\text{pc}(Q_1Z_1) \leq \text{pc}(QZ) \leq 1$ and $\text{pc}(Q_2Z_2) = 0$.

If it is an unfolding rule then there is a formula $\varphi \in \Phi$ that is being unfolded, and this φ is of the form $\varphi = \alpha \star \beta$ for some $\star \in \{\vee, \wedge, U, R\}$. Note that $\varphi \in \text{Cl}(\text{CTL})$ hence $\text{xc}(\varphi) \leq 3$ by Proposition 6.2, and since $5 \star \text{xc}(\alpha) + 5 \star \text{xc}(\beta) \leq \text{xc}(\varphi)$ we get $\text{xc}(\alpha) = 0$ and $\text{xc}(\beta) = 0$. If $\star \in \{\vee, \wedge\}$ then $\Phi_i \subseteq (\Phi \setminus \{\varphi\}) \cup \{\alpha, \beta\}$ for each i , hence $\text{pc}(QZ_i) \leq \text{pc}(QZ) \leq 1$. Otherwise $\star \in \{U, R\}$ and $\text{xc}(\varphi) = 1$, which means $\text{xc}(\psi) = 0$ for every $\psi \in \Phi \setminus \{\varphi\}$, and we have $\Phi_i \subseteq (\Phi \setminus \{\varphi\}) \cup \{\alpha, \beta, X\varphi\}$ for each i . Thus either $(\varphi, X\varphi) \in Z_i$ and $\text{pc}(QZ_i) = 1$ or $(\varphi, X\varphi) \notin Z_i$ and $\text{pc}(QZ_i) = 0$.

If it is a modal rule then all the formulas in Φ are X-formulas, but since $\text{pc}(QZ) \leq 1$ it must be that Φ contains exactly one formula, say $X\psi$. Now $Z_1 = \{(X\psi, \psi)\}$ hence the range of Z_1 is a singleton. This gives $\text{pc}(QZ_1) \leq 1$. \square

If the states of the automaton \mathbb{A}_ξ contain at most one path-formula, then the “lists” from \mathbb{B}_ξ contain at most one formula as well. This ensures that the clusters of $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ are simply rings of silent steps with one guarded step. Thus, when guardifying the automaton only one state per cluster remains. Also, note that for clusters of size 1 there is no need for priorities other than 0 or 1.

Theorem 6.5. *For every CTL_Σ formula there is an equivalent guarded modal Büchi automaton that has dominated clusters of size 1.*

Proof. Let $\xi \in \text{CTL}_\Sigma$. Note that any parity automaton with clusters of size 1 is essentially a Büchi automaton. Given Theorem 4.26, it is therefore enough to prove that $(\mathbb{A}_\xi \odot \mathbb{B}_\xi)^\gamma$ has clusters of size 1. This follows from the construction of Definition 4.21 once we have shown for all sequences $(a_0, b_0) \rightarrow_\circ (a_1, b_1) \rightarrow_\tau \dots \rightarrow_\tau (a_n, b_n) \rightarrow_\circ (a_{n+1}, b_{n+1})$ in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ that if (a_{n+1}, b_{n+1}) belongs to the same cluster as (a_1, b_1) , then $(a_{n+1}, b_{n+1}) = (a_1, b_1)$.

Now let $(a_0, b_0) \rightarrow_\circ (a_1, b_1) \rightarrow_\tau \dots \rightarrow_\tau (a_n, b_n) \rightarrow_\circ (a_{n+1}, b_{n+1})$ in $\mathbb{A}_\xi \odot \mathbb{B}_\xi$ such that (a_{n+1}, b_{n+1}) belongs to the same cluster as (a_1, b_1) . By Proposition 4.7 there are $Q, Z_1, \dots, Z_{n+1}, \Psi_1, \dots, \Psi_{n+1}$ such that $(a_i, b_i) = (QZ_i, Q\Psi_i)$ for $i > 0$. Since the rule for (a_0, b_0) is a modal rule, we know that the range of Z_0 contains only \mathbf{X} -formulas, and by Proposition 6.4 there can only be one such formula. Thus $Z_1 = \{(\mathbf{X}\varphi, \varphi)\}$ for some φ . From this it also follows that either $\Psi_1 = \emptyset$ or $\Psi_1 = \{\varphi\}$. Note that since $\mathbf{X}\varphi \in \text{Cl}(\text{CTL})$ it follows from Proposition 3.10 and Proposition 3.6 that $\varphi \in \text{CTL}$.

Suppose this φ is of the form $\alpha\mathbf{U}\beta$, then the applicable rule for QZ_1 must be the unfolding of φ and the resulting continuations will be subsets of $\{(\varphi, \alpha), (\varphi, \beta), (\varphi, \mathbf{X}\varphi)\}$. Since $\text{Cl}(\{\alpha, \beta\}) \subsetneq \text{Cl}(\{\varphi\})$, Proposition 4.7 tells us that Z_2 must contain $\{(\varphi, \mathbf{X}\varphi)\}$. Now if $Q = \mathbf{E}$ then $\Psi_2 = \{\varphi\}$ and if $Q = \mathbf{A}$ then $\Psi_2 = \emptyset$. For any $i \in \{2, \dots, n-1\}$ we have that the rule for $Q_i Z_i$ is not modal, hence $(\mathbf{X}\varphi, \mathbf{X}\varphi) \in Z_{i+1}$. Also Proposition 6.4 ensures that no other \mathbf{U} -formulas or \mathbf{R} -formulas can appear. Thus we will have $\text{Ran}(Z_n) = \{\mathbf{X}\varphi\}$ and either $Q = \mathbf{E}$ and $\Psi_n = \{\varphi\}$ or $Q = \mathbf{A}$ and $\Psi_n = \emptyset$. The rule for QZ_n is modal and we find $Z_{n+1} = \{(\mathbf{X}\varphi, \varphi)\} = Z_1$. Now we have either $Q = \mathbf{E}$ and $\Psi_{n+1} = \{\varphi\}$ or $Q = \mathbf{A}$ and $\Psi_{n+1} = \emptyset$. If $Q = \mathbf{A}$ then certainly $\Psi_1 \subseteq \{\varphi\} \cap \Phi_{\mathbf{R}} = \emptyset$. If $Q = \mathbf{E}$ then we have shown, without specifying Ψ_1 , that $\Psi_{n+1} = \{\varphi\}$. But since $(QZ_1, Q\Psi_1)$ and $(QZ_{n+1}, Q\Psi_{n+1})$ belong to the same cluster, this must mean that $\Psi_1 = \{\varphi\}$. This means $(QZ_1, Q\Psi_1) = (QZ_{n+1}, Q\Psi_{n+1})$, as desired.

The case where φ is of the form $\alpha\mathbf{R}\beta$ is similar. If φ would be of the form ℓ then QZ_1 would be terminal, which is impossible since $QZ_1 \rightarrow QZ_2$. If φ would be of the form $Q'\psi$, then we would have $Q'\psi \in \text{Cl}(\text{Ran}(Z_1)) \setminus \text{Cl}(\text{Ran}(Z_2))$ which contradicts Proposition 4.7. If φ would be of the form $\alpha \vee \beta$ or $\alpha \wedge \beta$, then since $\varphi \in \text{CTL}$ it must be that $\varphi \in \text{CTL}_\Sigma$, but then $\text{Ran}(Z_2) \subseteq \{\alpha, \beta\}$ and $\varphi \in \text{Cl}(\text{Ran}(Z_1)) \setminus \text{Cl}(\text{Ran}(Z_2))$, which again is impossible. \square

At the start of chapter 3 we saw that the unfolding of the CTL^* formula $\varphi\mathbf{U}\psi$ is $\psi \vee (\varphi \wedge \mathbf{X}(\varphi\mathbf{U}\psi))$. The construction of \mathbb{A}_ξ is based on this unfolding, and in fact the automaton $(\mathbb{A}_\xi \odot \mathbb{B}_\xi)^\gamma$ for $\xi := p\mathbf{U}q$ will have a state a with $\Delta(a) = q \vee (p \wedge \diamond a)$. Now note that when we restrict Proposition 3.17 to singleton clusters, all \exists -dominated states will have a transition term that is equivalent to one of the form $\Delta(a) = \psi \vee (\varphi \wedge \diamond a)$. If we can replace the $\text{gML}_1(A)$ terms φ and ψ with CTL_Σ formulas φ' and ψ' , then this suggests the formula $\varphi'\mathbf{U}\psi'$ as a translation for the state a . Indeed it is easy to turn modal formulas into CTL formulas, by replacing \diamond with \mathbf{EX} and \square with \mathbf{AX} . This idea is the basis for the translation from modal automata to CTL formulas.

Definition 6.6. Let $\mathbb{A} = (A, \Delta, \Omega, a_I)$ be a guarded modal Büchi automaton with dominated clusters of size 1. Using induction on cluster-depth, we create for every $a \in A$ a formula $\xi_a \in \text{CTL}_\Sigma$. Let $a \in A$, and suppose that ξ_b has been defined for every b with $a \rightarrow b$. Let C be the cluster that a belongs to, then $C = \{a\}$. Therefore Proposition 3.17 tells us that $\Delta(a)$ is equivalent to either $\alpha \vee (\beta \wedge \diamond a)$ or $\alpha \wedge (\beta \vee \square a)$, where α and β contain no occurrences of a . Together with $\Omega(a)$, this leads to four different types of states.

1. If $\Delta(a) \equiv \alpha \vee (\beta \wedge \diamond a)$ and $\Omega(a) = 1$, take $\xi_a := \mathbf{E}(\beta' \mathbf{U} \alpha')$.
2. If $\Delta(a) \equiv \alpha \vee (\beta \wedge \diamond a)$ and $\Omega(a) = 2$, take $\xi_a := \mathbf{E}(\alpha' \mathbf{R}(\alpha' \vee \beta'))$.
3. If $\Delta(a) \equiv \alpha \wedge (\beta \vee \square a)$ and $\Omega(a) = 2$, take $\xi_a := \mathbf{A}(\beta' \mathbf{R} \alpha')$.
4. If $\Delta(a) \equiv \alpha \wedge (\beta \vee \square a)$ and $\Omega(a) = 1$, take $\xi_a := \mathbf{A}(\alpha' \mathbf{U}(\alpha \wedge \beta'))$.

Here α' and β' are created from α and β by replacing every $\diamond b$ with $\mathbf{EX}\xi_b$ and every $\square b$ with $\mathbf{AX}\xi_b$. Note that $\alpha', \beta' \in \text{CTL}_\Sigma$. Once all ξ_a have been defined, construct the map $\cdot' : \text{gML}_1(A) \rightarrow \text{CTL}_\Sigma : \alpha \mapsto \alpha'$ by replacing every occurrence of $\diamond a$ with $\mathbf{EX}\xi_a$ and every occurrence of $\square a$ with $\mathbf{AX}\xi_a$. \triangleleft

First we need to show that replacing \diamond with \mathbf{EX} and \square with \mathbf{AX} works, hence that $\alpha' \equiv \alpha$ when α does not contain occurrences of the current state.

Proposition 6.7. *Let \mathbb{A} be a guarded modal Büchi automaton with dominated clusters of size 1. Let $a \in A$ and write $C := \{a\}$. Suppose that $\xi_b \equiv (\mathbb{A}, b)$ for all $b \in A$ with $a \rightarrow b$. Let \mathbb{S} be a transition system. For every $\pi \in \Pi(\mathbb{S})$, every $i \in \mathbb{N}$ and every $\varphi \in \text{Sb}(\Delta(a))$ that does not contain occurrences of a , if a player has a winning strategy for $\mathcal{E}(\varphi', \mathbb{S})@(\pi(i), \varphi')$ then that player has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(i, \varphi)$.*

Proof. Let $\pi \in \Pi(\mathbb{S})$, $i \in \mathbb{N}$ and $\psi \in \text{Sb}(\Delta(a))$. Define $s := \pi(i)$. Suppose that \exists has a winning strategy for $\mathcal{E}(\psi', \mathbb{S})@(s, \psi')$; the proof for the case where \forall has a winning strategy is dual. By Proposition 3.14 we may assume that f is positional. As \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(i, \psi)$, \exists plays a shadow match of $\mathcal{E}(\psi', \mathbb{S})@(s, \psi')$ that is consistent with f , such that real positions (i, φ) correspond to shadow positions (s, φ') . If $\varphi = \varphi_1 \vee \varphi_2$ then $\varphi' = \varphi'_1 \vee \varphi'_2$; here f tells \exists to choose some (s, φ'_i) and in the real match \exists should choose (i, φ_i) . If $\varphi = \varphi_1 \wedge \varphi_2$ then $\varphi' = \varphi'_1 \wedge \varphi'_2$, and any move by \forall to (i, φ_i) can be mimicked with the move to (s, φ'_i) in the shadow match. In this way both matches are played until, after a finite number of moves on both sides, positions (i, φ) and (s, φ') are reached where φ is one of ℓ , $\diamond b$ or $\square b$.

If $\varphi = \ell$ then $\varphi' = \ell$, and \exists wins both games if $\mathbb{S}, s \Vdash \ell$ and loses both if $\mathbb{S}, s \not\Vdash \ell$; because the shadow match is consistent with \exists 's winning strategy f , \exists wins. If $\varphi = \diamond b$ then $\varphi' = \mathbf{EX}\xi_b$. Here f tells \exists to move to $(\rho, \mathbf{X}\xi_b)$ for some $\rho \in \Pi(\mathbb{S}, s)$, and the shadow match then necessarily continues to (ρ^1, ξ_b) . Because $\xi_b \in \text{CTL}_\Sigma$, it then continues to $(\rho(1), \xi_b)$. Note that $sR\rho(1)$ and $b \notin C$, thus \exists can choose $(\rho(1), b)$ in the real game. Because the shadow match is consistent with f we have that $(\rho(1), \xi_b)$ is a winning position for \exists ,

hence $\mathbb{S}, \rho(1) \Vdash \xi_b$. Since $a \rightarrow b$, we get $\mathbb{S}, \rho(1) \Vdash (\mathbb{A}, b)$. This means that $(\rho(1), b)$ is a winning position for \exists , hence \exists wins the real match. If $\varphi = \Box b$ then $\varphi' = \mathbf{AX}\xi_b$. Suppose \forall moves to (t, b) where sRt . By seriality there is some $\rho' \in \Pi(\mathbb{S}, t)$, so define $\rho \in \Pi(\mathbb{S}, s)$ by taking $\rho(0) := s$ and $\rho(k+1) := \rho'(k)$. Let shadow- \forall move to $(\rho, \mathbf{X}\xi_b)$, then this necessarily continues first to $(\rho^1, \xi_b) = (\rho', \xi_b)$ and then to $(\rho'(0), \xi_b) = (t, \xi_b)$. These moves are consistent with f , so this is a winning position for \exists and therefore $\mathbb{S}, t \Vdash \xi_b$. Now because $a \rightarrow b$ we get $\mathbb{S}, t \Vdash (\mathbb{A}, b)$, thus (t, b) must be winning for \exists , hence \exists wins the real match. \square

This leaves us to prove that this translation works. The proof is based on the case distinction from Definition 6.6.

Theorem 6.8. *For every guarded modal Büchi automaton with dominated clusters of size 1 there is an equivalent CTL_Σ formula.*

Proof. Let $\mathbb{A} = (A, \Delta, \Omega, a_1)$ be a guarded modal Büchi automaton with dominated clusters of size 1. Using induction on cluster-depth, we show that $(\mathbb{A}, a) \equiv \xi_a$ for every $a \in A$. This then gives $\mathbb{A} \equiv \xi_{a_1}$. Let $a \in A$ and write $C := \{a\}$. Let \mathbb{S} be a transition system, then we need to show that $\mathbb{S}, s \Vdash \xi_a$ iff $\mathbb{S}, s \Vdash (\mathbb{A}, a)$ for all $s \in S$. We apply a case distinction based on ξ_a as defined in Definition 6.6.

Suppose (1.), thus $\xi_a = \mathbf{E}(\beta' \mathbf{U} \alpha')$. Let $\psi := \beta' \mathbf{U} \alpha'$. Let $s \in S$ and suppose that $\mathbb{S}, s \Vdash \xi_a$, then \exists has a winning strategy f for the game $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$. First, this strategy takes her to a position (π, ψ) for some $\pi \in \Pi(\mathbb{S}, s)$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$, \exists plays a shadow match of $\mathcal{E}(\xi_a, \mathbb{S})@(\pi, \psi)$. Whenever the real match is at (i, a) and the shadow match is at (π^i, ψ) , they necessarily continue to $(i, \alpha \vee (\beta \wedge \Diamond a))$ and $(\pi^i, \alpha' \vee (\beta' \wedge \mathbf{X}\psi))$ respectively. Here if f instructs \exists to choose (π^i, α') , then she should move to (i, α) in the real game; otherwise it instructs her to choose $(\pi^i, \beta' \wedge \mathbf{X}\psi)$ and she should move to $(i, \beta \wedge \Diamond a)$. From there an \forall -move to (i, β) can be mimicked with a shadow move to (π^i, β') , and an \forall -move to $(i, \Diamond a)$ can be mimicked with a shadow move to $(\pi^i, \mathbf{X}\psi)$. Note that if the matches are at $(i, \Diamond a)$ and $(\pi^i, \mathbf{X}\psi)$, then they necessarily continue to $(i+1, a)$ and (π^{i+1}, ψ) respectively. It cannot be that both matches go through their respective positions (i, a) and (π^i, ψ) infinitely often since then \exists would lose the shadow match, which is consistent with f . This would contradict the fact that f is a winning strategy. If the matches eventually reach positions (i, α) and (π^i, α') , then the shadow match is consistent with f hence (π^i, α') is winning for \exists . Because $\alpha' \in \text{CTL}_\Sigma$, a match from (π^i, α') necessarily continues to $(\pi(i), \alpha')$, which must therefore also be winning for \exists . By Proposition 6.7 this means (i, α) is winning for \exists , and thus \exists wins the real match. If not, then the matches must eventually reach positions (i, β) and (π^i, β') , and it similarly follows that (i, β) is winning for \exists and that \exists wins the real match. Either way, we conclude that $(0, a)$ is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. Now by Proposition 3.20 this means that (s, a) is a winning position in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})$, and thus $\mathbb{S}, s \Vdash (\mathbb{A}, a)$.

Now let $s \in S$ and suppose that $\mathbb{S}, s \Vdash (\mathbb{A}, a)$. By Proposition 3.20 there is $\pi \in \Pi(\mathbb{S}, s)$ such that $(0, a)$ is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. While \exists and \forall play a match of $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$, \exists plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. The first move

for \exists should be (π, ψ) . Now as above the two matches are played very similarly, with positions (π^i, ψ) corresponding to shadow positions (i, a) . Again, it cannot be that both matches are infinite, for then a would be visited infinitely often while $\Omega(a) = 1$. Thus both matches must eventually reach positions (π^i, φ') and (i, φ) where φ is either α or β . The shadow position (i, φ) is a winning position of \exists . If it were the case (π^i, φ') is not a winning position for \exists , then by Proposition 3.14 this means it is a winning position for \forall . By Proposition 6.7 this would mean that (i, φ) is winning for \forall , which contradicts that it is winning for \exists . Therefore (π^i, φ') must be a winning position for \exists and she wins the real match. Thus we have $\mathbb{S}, s \Vdash \xi_a$.

Suppose (2.), thus $\xi_a = \mathbf{E}(\alpha'R(\alpha' \vee \beta'))$. Let $\psi := \alpha'R(\alpha' \vee \beta')$. Let $s \in S$ and suppose that $\mathbb{S}, s \Vdash \xi_a$, then \exists has a winning strategy f for the game $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$. First, this strategy takes her to a position (π, ψ) for some $\pi \in \Pi(\mathbb{S}, s)$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$, \exists plays a shadow match of $\mathcal{E}(\xi_a, \mathbb{S})@(\pi, \psi)$. Whenever the real match is at (i, a) and the shadow match is at (π^i, ψ) , they necessarily continue to $(i, \alpha \vee (\beta \wedge \diamond a))$ and $(\pi^i, (\alpha' \vee \beta') \wedge (\alpha' \vee X\psi))$ respectively. If \forall moves to $(\pi^i, \alpha' \vee \beta')$, then f instructs \exists to choose either (π^i, α') or (π^i, β') . Now \exists should choose (i, α) if the former or $(i, \beta \wedge \diamond a)$ if the latter, and in the latter case shadow- \forall can be made to choose (i, β) . If \forall instead moves to $(\pi^i, \alpha' \vee X\psi)$, then f instructs \exists to choose either (π^i, α') or $(\pi^i, X\psi)$. Again \exists should choose (i, α) or $(i, \beta \wedge \diamond a)$ accordingly, and in the latter case shadow- \forall chooses $(i, \diamond a)$. Note that if the matches are at $(i, \diamond a)$ and $(\pi^i, X\psi)$, then they necessarily continue to $(i+1, a)$ and (π^{i+1}, ψ) respectively. As in case (1.), if the matches ever reach (i, α) and (π^i, α') or (i, β) and (π^i, β') , then Proposition 6.7 gives us that \exists wins both matches. If they never reach such positions, then both matches are infinite. The only state in A which is visited infinitely often is a ; since $\Omega(a) = 2$ we get that \exists wins the real match and therefore $\mathbb{S}, s \Vdash (\mathbb{A}, a)$.

Now let $s \in S$ and suppose that $\mathbb{S}, s \Vdash (\mathbb{A}, a)$. By Proposition 3.20 there is $\pi \in \Pi(\mathbb{S}, s)$ such that $(0, a)$ is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. While \exists and \forall play a match of $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$, \exists plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. The first move for \exists should be (π, ψ) . Once again the two matches are played very similarly, with positions (π^i, ψ) corresponding to shadow positions (i, a) . If both matches are infinite, then the formula ψ occurs infinitely often, and ψ is an \mathbf{R} -formula. Thus \exists wins the real match, and this means $\mathbb{S}, s \Vdash \xi_a$.

Suppose (3.), thus $\xi_a = \mathbf{A}(\beta'R\alpha')$. Let $\psi := \beta'R\alpha'$. Let $s \in S$ and suppose that $\mathbb{S}, s \not\Vdash \xi_a$, then by Proposition 3.14 \forall has a winning strategy f for the game $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$. First, this strategy takes him to a position (π, ψ) for some $\pi \in \Pi(\mathbb{S}, s)$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$, \forall plays a shadow match of $\mathcal{E}(\xi_a, \mathbb{S})@(\pi, \psi)$. As a dual to (1.), we find that \forall wins both matches. Again by Proposition 3.14, this means that $\mathbb{S}, s \not\Vdash (\mathbb{A}, a)$.

Now let $s \in S$ and suppose that $\mathbb{S}, s \not\Vdash (\mathbb{A}, a)$. By Proposition 2.43 this means \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$, and by Proposition 3.20 there is $\pi \in \Pi(\mathbb{S}, s)$ such that $(0, a)$ is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. While \exists and \forall play a match of $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$, \forall plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. The first move for \forall should be (π, ψ) . As a dual to (1.), we find that \forall wins both matches. Again by

Proposition 3.14, this means that $\mathbb{S}, s \not\ll \xi_a$.

Suppose (4.), thus $\xi_a = \mathbf{A}(\alpha' \mathbf{U}(\alpha' \wedge \beta'))$. Let $\psi := \alpha' \mathbf{U}(\alpha' \wedge \beta')$. Let $s \in S$ and suppose that $\mathbb{S}, s \not\ll \xi_a$, then by Proposition 3.14 \forall has a winning strategy f for the game $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$. First, this strategy takes him to a position (π, ψ) for some $\pi \in \Pi(\mathbb{S}, s)$. While \exists and \forall play a match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$, \exists plays a shadow match of $\mathcal{E}(\xi_a, \mathbb{S})@(\pi, \psi)$. As a dual to (2.), we find that \forall wins both matches. Again by Proposition 3.14, this means that $\mathbb{S}, s \not\ll (\mathbb{A}, a)$.

Now let $s \in S$ and suppose that $\mathbb{S}, s \not\ll (\mathbb{A}, a)$. By Proposition 2.43 this means \forall has a winning strategy for $\mathcal{A}(\mathbb{A}, \mathbb{S})@(s, a)$, and by Proposition 3.20 there is $\pi \in \Pi(\mathbb{S}, s)$ such that $(0, a)$ is a winning position for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$. While \exists and \forall play a match of $\mathcal{E}(\xi_a, \mathbb{S})@(s, \xi_a)$, \forall plays a shadow match of $\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)@(0, a)$. The first move for \forall should be (π, ψ) . As a dual to (2.), we find that \forall wins both matches. Again by Proposition 3.14, this means that $\mathbb{S}, s \not\ll \xi_a$. \square

6.2 CTL as a fragment of the modal μ -calculus

In the previous section we obtained a characterization of CTL as a class of modal automata with dominated clusters, by restricting the size of the clusters to 1. In this section we will do the same with $\mu_{\text{dom}}\text{ML}$ by looking at the one-variable fragment. Because we work with clean formulas we cannot really restrict the number of variables to 1; we will instead demand that fixpoint subformulas don't have free variables.

Definition 6.9. Let $\mu\text{ML}[1]$ denote the fragment of μML containing those formulas for which no fixpoint subformulas have free variables. \triangleleft

We write $\mu_1\text{ML}[1]$ for the intersection of $\mu_1\text{ML}$ and $\mu\text{ML}[1]$, and $\mu_{\text{dom}}\text{ML}[1]$ for the intersection of $\mu_{\text{dom}}\text{ML}$ and $\mu\text{ML}[1]$. Note that although the fragment $\mu_1\text{ML}$ is just as expressive as μML , this does not hold when restricting the variables that can be used. As a particular example, the $\mu\text{ML}[1]$ formula $\mu x.(p \wedge \diamond \diamond x)$ has typical fixpoints but it is inexpressible in $\mu_{\text{dom}}\text{ML}[1]$. In order to see that $\mu\text{ML}[1]$ indeed corresponds to a fragment of μML where only one variable is used, it is enough to realize that \bowtie_ξ -classes can contain at most one bound variable.

Proposition 6.10. *Let ξ be a $\mu\text{ML}[1]$ formula. Every non-trivial \bowtie_ξ -class contains exactly one fixpoint formula.*

Proof. Because bound variables are the only formulas that have non-subformula derivatives, any \bowtie_ξ equivalence class must contain at least one bound variable in order to be non-trivial. Let $x \in \text{BV}(\xi)$ be part of a non-trivial \bowtie_ξ equivalence class Φ . There is $\varphi \in \Phi \setminus \{x\}$ with $\varphi \in \nabla_\xi(x)$ and this φ can only be $\lambda_x x.\vartheta_x$ and thus $\lambda_x x.\vartheta_x \in \Phi$. Let $\lambda_y y.\vartheta_y \in \Phi$. Because ξ is a $\mu\text{ML}[1]$ formula, $\text{FV}(\lambda_x x.\vartheta_x) = \emptyset$ and $\text{FV}(\lambda_y y.\vartheta_y) = \emptyset$. By Proposition 5.14 this means that $\lambda_y y.\vartheta_y$ must be a subformula of $\lambda_x x.\vartheta_x$ and that $\lambda_x x.\vartheta_x$ must be a subformula of $\lambda_y y.\vartheta_y$. This is only possible if $\lambda_x x.\vartheta_x = \lambda_y y.\vartheta_y$. \square

To show that the logics CTL^* and $\mu_{\text{dom}}\text{ML}[1]$ have the same expressive power, we can use the translations between modal automata that we already have. The logic $\mu_1\text{ML}[1]$ corresponds very neatly to the class of guarded modal automata with singleton clusters, and dominance carries over.

Theorem 6.11. *For every $\mu_{\text{dom}}\text{ML}[1]$ formula there exists an equivalent guarded modal parity automaton with dominated clusters of size 1.*

Proof. Let $\xi \in \mu_{\text{dom}}\text{ML}[1]$ and let \mathbb{A}_ξ be the guarded modal parity automaton as constructed in Definition 5.4. By Proposition 5.15 this automaton has dominated clusters. By Proposition 5.6 and Proposition 6.10, the clusters of \mathbb{A}_ξ are of size 1. \square

Theorem 6.12. *For every guarded modal parity automaton with dominated clusters of size 1 there exists an equivalent $\mu_{\text{dom}}\text{ML}[1]$ formula.*

Proof. Let \mathbb{A} be a guarded modal parity automaton with dominated clusters of size 1 and let $\xi_{\mathbb{A}}$ be the guarded modal parity automaton as constructed in Definition 5.8. By Proposition 5.12 this formula has typical fixpoints. Let φ be a fixpoint subformula of $\xi_{\mathbb{A}}$, then there is a cluster C and a set $I \in \mathcal{I}$ such that $\varphi = \lambda_I x_I. \vartheta_I$. Because C has size 1, $\mathcal{I} = \{\{1\}\}$ and thus $I = \{1\}$. Now $\text{FV}(\vartheta_I) = \{x_I\}$ and thus $\text{FV}(\varphi) = \emptyset$. This shows that $\xi_{\mathbb{A}} \in \mu_{\text{dom}}\text{ML}[1]$. \square

Theorem 6.13. *The logics CTL and $\mu_{\text{dom}}\text{ML}[1]$ are equally expressive.*

Proof. By Theorem 6.5, Theorem 6.8, Theorem 6.11 and Theorem 6.12. \square

Chapter 7

Conclusion

In this paper we gave evaluation game semantics for CTL^* and we introduced a class of modal parity automata that CTL^* formulas can be translated into. We characterized this class of automata as a fragment of the modal μ -calculus. The class of automata in question is based on a syntactic property we called ‘dominance’, and it carries over in a natural way when going from modal automata to μML formulas and back.

We also showed that for a guarded modal parity automaton with dominated clusters, the Mostowski index can be reduced to be at most 2. In particular, \exists -dominated and \forall -dominated clusters can be given Büchi and co-Büchi acceptance conditions respectively. From a μML perspective, this means that the fragment consisting of μML formulas with dominated fixpoints is equivalent to that fragment up to alternation depth 2. This is not entirely surprising since Dam (1990) translates CTL^* into a fragment of μML that has alternation depth 2.

Lastly, we gave a complete characterization of CTL as a class of modal parity automata by restricting to clusters of size 1, and as a fragment of μML by restricting to the one-variable fragment of $\mu_{\text{dom}}\text{ML}$. This illustrates the simplicity of CTL , or conversely the relative strength of the one-variable modal μ -calculus.

Discussion

First a note on complexity. The automaton construction in chapter 4 is exponential in size, as its underlying set of states is based on the powerset of the closure. On the other hand the guardification given by Definition 4.21 does not increase the size of the automaton. As remarked by Lenzi (2005), “any translation [from CTL^* to μML] must be exponentially complicated, because the modal μ -calculus is decidable in exponential time, whereas CTL^* is decidable in no less than double exponential time.” It was shown by Vardi and Wolper (1985) that the satisfiability problem for CTL^* is logspace hard for deterministic double exponential time, and then by Emerson and Halpern (1999) that the satisfiability problem for μML can be solved in deterministic exponential time. Because modal automata are so closely related to μML formulas, a non-exponential translation would be impossible.

The class of automata that CTL* is translated into, i.e. guarded modal parity automata with \exists -dominated Büchi clusters and \forall -dominated co-Büchi clusters, is currently too broad to characterize CTL*. This is given by the following proposition, together with Theorem 5.24 and Theorem 5.23. Note that $\nu x.(p \wedge \diamond \diamond x) \equiv \nu x.(p \wedge \diamond \nu y.\diamond x) \in \mu_{\text{dom}}\text{ML}$.

Proposition 7.1. *The formula $\nu x.(p \wedge \diamond \diamond x)$ is not expressible in CTL*.*

This can be proven using Ehrenfeucht-Fraïssé games for CTL*. Alternatively, if we consider only linear transition systems over $\{p\}$, which can be seen as $\wp(\{p\})$ -streams, then $\nu x.(p \wedge \diamond \diamond x)$ denotes the language L that contains all streams for which every even-numbered symbol is $\{p\}$. On linear models CTL* and LTL coincide, which is equivalent to first-order logic as shown by Kamp (1968). McNaughton and Papert (1971) show that first-order logic on linear models corresponds with the class of star-free regular languages – that is, regular languages that can be constructed without the use of Kleene-star but with the use of complementation. The language L is not definable as a star-free language, hence cannot be expressible in CTL*.

Diekert and Gastin (2008) show that LTL corresponds to the class of nondeterministic Büchi stream automata that are *counter-free*; given \rightarrow_w as defined in Definition 2.33, a nondeterministic stream automaton \mathbb{B} is counter-free if $b \rightarrow_{w^m} b$ implies $b \rightarrow_w b$ for all $b \in B$, all words w and all $m \geq 1$. If we can view a guarded modal automata with dominated clusters as a $\wp(\text{Prop})$ -stream automaton, we can apply the notion of counter-freeness to it. By Proposition 3.17 the transition term of an \exists -dominated state a is a disjunction where each disjunct is of the form $\varphi \wedge \diamond a'$. If we treat subformulas $\diamond b$ and $\Box b$ where $a \rightarrow b$ as extra propositional letters, then every such formula φ has a propositional extension $\llbracket \varphi \rrbracket \in \wp(\text{Prop})$. Now we can write $a \rightarrow_{\llbracket \varphi \rrbracket} a'$ when the disjunct $\varphi \wedge \diamond a'$ occurs in the transition term of a . The current constructed automata are not immediately counter-free, but this might be obtained by changing the Dam rules for U and R formulas. Currently the E-rule for the unfolding of a U-formula is based on $\alpha \text{U} \beta \equiv \beta \vee (\alpha \wedge \text{X}(\alpha \text{U} \beta))$, but it could also be based on $\alpha \text{U} \beta \equiv \beta \vee (\alpha \wedge \neg \beta \wedge \text{X}(\alpha \text{U} \beta))$ where $\neg \beta$ is as defined in Definition 2.8. The resulting modal automaton might be counter-free when seen as a $\wp(\text{Prop})$ -automaton. A better understanding of counter-free behaviour in modal automata might lead to a complete characterization of CTL* as a class of automata.

Even then, there is something inherently tricky about creating CTL* formulas for modal automata with non-singleton clusters. As a specific example, take $\mathbb{A} = (A, \Delta, \Omega, a)$ where $A = \{a, b\}$, $\Delta(a) = \Delta(b) = \diamond b \vee (p \wedge \diamond a)$, $\Omega(a) = 2$ and $\Omega(b) = 1$; note that \mathbb{A} is a guarded modal parity automaton and that its only cluster, $\{a, b\}$, is \exists -dominated. During an acceptance match of \mathbb{A} , it is certainly possible to visit both states a and b infinitely often. A seemingly reasonable approach to creating formulas for \mathbb{A} in some logic is to create formulas φ_a and φ_b such that an evaluation match visits φ_a whenever the acceptance match visits a and visits φ_b whenever the acceptance match visits b . For this to be possible, we would need $\varphi_a \bowtie \varphi_b$.

Translating \mathbb{A} to μML , we could first give $\xi'_a = \nu x_a.(\diamond x_b \vee (p \wedge \diamond x_a)) \in \mu\text{ML}(\{x_b\})$ and $\xi'_b = \mu x_b.(\diamond x_b \vee (p \wedge \diamond x_a)) \in \mu\text{ML}(\{x_a\})$. We can see this as a system of equations in two variables x_a and x_b , and the solution to this system is $\xi_a = \nu x_a.(\diamond \xi'_b \vee (p \wedge \diamond x_a)) \in \mu\text{ML}(\emptyset)$,

which is obtained by replacing x_b in ξ'_a with ξ'_b . Note that $\xi_a \bowtie \xi'_b$ because ξ'_b contains x_a as a variable that is bound by ξ_a , so that an evaluation match for ξ_a can go from ξ_a to ξ'_b and then via x_a back to ξ_a . Solving a system of equations is essentially what is done in the construction as given in Definition 5.8.

In a similar attempt for CTL^* , we could first create the formulas $\psi'_a = (\mathbf{X}x_b)\mathbf{R}(\mathbf{X}x_b \vee p)$ and $\psi'_b = \top\mathbf{U}(p \wedge \mathbf{X}x_a)$ if we allow x_a and x_b as extra proposition letters that we will later replace by CTL^* formulas. However, when creating $\psi_a = (\mathbf{X}\psi'_b)\mathbf{R}(\mathbf{X}\psi'_b \vee p)$ by replacing the variable x_b in ψ'_a with ψ'_b , the variable x_a remains free in ψ_a , because nothing in CTL^* can bind it. In fact, there is no CTL^* formula χ that we can replace x_a with in order to get $\psi_a \triangleright \psi'_b \triangleright \chi \triangleright \psi_a$, because the \bowtie equivalence class of the \mathbf{R} -formula ψ'_a cannot contain the \mathbf{U} -formula ψ_b , as stated by Proposition 3.6. In this light CTL^* seems to be unable to solve systems of equations with more than one variable, and a completely different approach would have to be taken to find the CTL^* formula that corresponds to \mathbb{A} , which is $\text{EGF}p$.

Open questions

First and foremost, analyzing and sharpening the automaton construction given in this paper might lead to complete characterization of CTL^* as a class of automata. It might be worth exploring the notion of counter-free modal automata as discussed above.

When given the class of automata that corresponds to CTL^* , the class that corresponds to CTL is obtained by restricting the clusters to be of size 1. If we can detect when it is possible for the automaton generated by a CTL^* formula to have singleton clusters and when this is impossible, this might lead to an effective characterization of the CTL^* formulas that are also expressible in CTL .

Moller and Rabinovich (2001) show that CTL^* corresponds to the bisimulation invariant fragment of Monadic Path Logic on trees. It should be possible to reprove those results by giving effective translations between MPL and the class of automata that characterizes CTL^* . Furthermore, restricting these translations to automata with singleton clusters might expose a fragment of MPL that corresponds to CTL .

Bibliography

- F. Carreiro and Y. Venema. PDL inside the μ -calculus, a syntactic and an automata-theoretic characterization. *Advances in Modal Logic*, 10, 2014.
- E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. *Proceedings of the Logic of Programs Workshop*, pages 52–71, 1981.
- M. Dam. Translating CTL* into the modal μ -calculus. *ECS-LFCS*, 90(123), 1990.
- V. Diekert and P. Gastin. First-order definable languages. *Logic and Automata: History and Perspectives, Texts in Logic and Games*, pages 261–306, 2008.
- A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(2):129–141, 1961.
- E. Emerson and J. Halpern. “sometimes” and “not never” revisited: On branching versus linear time temporal logic. *Journal of the ACM - The MIT Press scientific computation series*, 33(1):151–178, 1986.
- E. Emerson and J. Halpern. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29(1):132–158, 1999.
- E. Emerson and C. Jutla. Tree automata, mu-calculus and determinacy. *Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 368–377, 1991.
- D. Janin and I. Walukiewicz. Automata for the modal μ -calculus and related results. *Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science*, pages 552–562, 1995.
- D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ -calculus with respect to monadic second order logic. *Proceedings of the 7th International Conference on Concurrency Theory*, pages 263–277, 1996.
- J. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles (California), 1968.

- S. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, pages 3–42, 1951.
- D. Kozen. Results on the propositional μ -calculus. *Theoretical Computer Science*, 27(3): 333–354, 1983.
- L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, 3(2):125–143, 1977.
- G. Lenzi. The modal mu-calculus, a survey. *Task quarterly*, 9(3):293–316, 2005.
- R. McNaughton and S. Papert. *Counter-free*. The MIT Press, Cambridge, Massachusetts, 1971.
- F. Moller and A. Rabinovich. Counting on CTL*: on the expressive power of monadic path logic. *Information and Computation*, 184(1):147–159, 2001.
- A. Pnueli. The temporal logic of programs. *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, 1977.
- V. Pratt. A decidable μ -calculus. *22nd Annual Symposium on Foundations of Computer Science*, pages 412 – 427, 1981.
- A. Prior. Time and modality. *Philosophy*, 34(128), 1957.
- R. Fraïssé. Sur quelques classifications des systèmes de relations. *Publications scientifiques de l'Université d'Alger*, 1955.
- M. Vardi and P. Wolper. Improved upper and lower bounds for modal logics of programs. *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 240–251, 1985.
- I. Walukiewicz. Monadic second order logic on tree-like structures. *13th Annual Symposium on Theoretical Aspects of Computer Science*, 1046:401–414, 1996.
- W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.

Index

Symbols

\mathbb{A}	15	\rightsquigarrow	55
(\mathbb{A}, a)	15	c_I	52
$\mathbb{A}^{\dot{\div}}$	57	$\text{Cl}(\Phi)$	20
\mathbb{A}^{\downarrow}	56	$\text{Cl}(\varphi)$	20
\mathbb{A}^{γ}	45	$\text{Cl}_{\Pi}(\Phi)$	20
\mathbb{A}_{ξ} (for CTL^*)	34	$\text{Cl}_{\Sigma}(\Phi)$	20
\mathbb{A}_{ξ} (for $\mu_1\text{ML}$)	50	CTL	6
$\mathbb{A}_{\xi} \odot \mathbb{B}_{\xi}$	44	CTL_{Π}	6
Acc	34	CTL_{Σ}	6
$\mathcal{A}(\mathbb{A}, \mathbb{S})$	15	CTL^*	6
$\mathcal{A}(\mathbb{A}, \mathbb{S}, C, \pi)$	27	CTL_{Π}^*	6
\forall	8	CTL_{Σ}^*	6
\forall -dominated (for ML_1)	26	δ	14
\forall -dominated (for μML)	53	Δ	15
\wedge	6	\diamond	10
\mathbf{A}	6	\diamond -guarded	15
\mathbf{A} -cluster	35	ℓ	5
A_{ξ} (for CTL^*)	34	ϵ	31
A_{ξ} (for $\mu_1\text{ML}$)	50	\equiv	7
\mathbb{B}	14	\mathbf{E}	6
\mathbb{B}_{ξ}	42	\mathbf{E} -cluster	35
\square	10	$\mathcal{E}(\varphi, \mathbb{S})$ (for CTL^*)	23
\square -guarded	15	$\mathcal{E}(\varphi, \mathbb{S})$ (for μML)	12
$\text{BV}(\varphi)$	10	\exists	8
B_{ξ}	42	\exists -dominated (for ML_1)	26
\rightarrow	16	\exists -dominated (for μML)	53
\rightarrow	16	\perp	5
\rightarrow_{σ}	14	$\text{fd}(\varphi)$	13
\rightarrow_{\circ}	45	φ_I	52
\rightarrow_{τ}	45	\mathbf{F}	6
\twoheadrightarrow	16	$\text{FV}(\varphi)$	10
\twoheadrightarrow_w	14	$\mathcal{G}@b$	9
		$\text{gML}_1(A)$	15

G	6
ic(QZ)	36
ic(φ)	36
\mathcal{I}	52
$I(j)$	52
λ	10
λ_I	52
λ_x	10
$\lambda_\xi(x)$	10
\prec_ξ	10
\preceq_ξ	10
Lit	5
ML(A)	48
ML ₁ (A)	15
μ ML	10
μ ML(Y)	10
μ ML _{Λ}	49
μ ML _{M}	49
μ ML[1]	67
μ_1 ML	49
μ_1 ML(Y)	49
μ_1 ML _{Λ} (Y)	49
μ_1 ML _{M} (Y)	49
μ_{dom} ML	53
μ	10
$\nabla(\varphi)$	20
$\nabla_\xi(\varphi)$	11
\neg (for CTL*)	7
\neg (for μ ML)	13
ν	10
Ω	8
$\Omega_\xi(x)$	13
$\Omega_\xi(\underline{x})$	50
\vee	6
pc(QZ)	62
π	6
π^k	6
$\Pi(\mathbb{S})$	6
$\Pi(\mathbb{S}, s)$	6
Prop	5
R	5
$r(I)$	52
R	6

R-cell	22
\mathbb{S}	5
S	5
\Vdash (for automata)	16
\Vdash (for CTL*)	7
\Vdash (for μ ML)	12
\Vdash_V	5
Sb(φ)	11
\triangleleft	21
\triangleleft_ξ	11
ϑ_I	52
ϑ_x	10
\boxtimes	21
\boxtimes_ξ	11
\top	5
U	6
U-cell	22
V	5
Var	10
Win \forall (\mathcal{G})	9
Win \exists (\mathcal{G})	9
xc(φ)	61
ξ_Δ (for CTL)	63
ξ_Δ (for μ_1 ML)	51
X	6
$\mathcal{Z}(\Phi)$	34

A

acceptance condition	15
acceptance game	15
accepting run	14
admissible move	8
applicable rule	34

B

Büchi cluster	17
binding fixpoint formula	10
bound variable	10

C

clean formula	10
closed under derivatives	11
closure	20
cluster	17

cluster-depth	17
cluster-path acceptance game	27
co-Büchi cluster	17
cofinal trace	22
color	8
colored game board	8
coloring	8
complete match	8
complete trace	11
consistent	9
continuation	34
D	
Dam rule	32
Dam term	32
decomposition	49
derivative (for μML)	11
derivative (for CTL^*)	20
descend	16
determinacy	9
deterministic stream automaton	14
direct trace	11
dominated cluster	26
dominated fixpoints	53
dominated formula (for ML_1)	26
dominated formula (for μML)	53
E	
evaluation game (for CTL^*)	23
evaluation game (for μML)	12
explosive complexity	61
extraction rule	32
F	
fixpoint operator	10
fixpoint-depth	13
free variable	10
G	
generalized cluster	17
guarded	15
guarded modal automaton	15
guarded modal one-step formula	15

I	
immediate complexity	36
induced sequence	8
initialized game	9
L	
literal	5
logical equivalence	7
logical complement (for CTL^*)	7
logical complement (for μML)	13
M	
match	8
modal automaton	15
modal Muller automaton	15
modal parity automaton	15
modal rule	32
modal one-step formula	15
Mostowski index	17
Muller condition	8
Muller game	8
N	
non-terminal rule	32
non-terminale state	34
O	
one-variable fragment	67
P	
parity game	8
partial match	8
path	6
path-formula	6
path-formula count	62
position	8
positional determinacy	9
positional strategy	9
priority function	8
proposition letter	5
Q	
quantified continuation	34
R	
reach	16

relational semantics	7
run through	34
run	14
S	
serial	5
silent-cycle safe	45
silent-priority safe	46
silent-step automaton	15
stalling trace	11
state-formula	6
state (of a modal automaton)	15
state (of a transition system)	5
strategy	9
stream	14
stream automaton	14
T	
temporal operator	6

terminal rule	32
terminal state	34
trace (for CTL*)	21
trace (for μ ML)	11
transition system	5
transition term	15
trivial cluster	17
U	
unfolding rule	32
unguarded	15
V	
valuation	5
variable	10
W	
winning position	9
winning strategy	9